Hierarchical Reinforcement Learning With Attention Reward

Extended abstract

Sihong Luo
State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications
Beijing, China
arphara@bupt.edu.cn

Jinghao Chen
State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications
Beijing, China
chenjh@bupt.edu.cn

Zheng Hu*
State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications
Beijing, China
huzheng@bupt.edu.cn

Chunhong Zhang
State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications
Beijing, China
zhangch@bupt.edu.cn

Benhui Zhuang
State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications
Beijing, China
zhuangbenhui@bupt.edu.cn

ABSTRACT

Hierarchical Reinforcement Learning (HRL) is a promising approach for complex tasks with greater sample efficiency because it can break a task into sets of short subtasks and provide a denser subgoal-related intrinsic reward, making credit assignments less challenging. However, none of the conventional subgoal-related intrinsic rewards utilize task-specified knowledge, which limits the sample efficiency of these HRL methods. We propose Hierarchical Reinforcement Learning with Attention Reward (HiAR) that motivates agents to focus on the part of the environment controlled by their actions. We introduce a measure of the control over each dimension in the state space and discuss how we integrated it into the HRL method to improve the sample efficiency.

KEYWORDS

Hierarchical reinforcement learning; intrinsic reward; contingency awareness

ACM Reference Format:
Sihong Luo, Jinghao Chen, Zheng Hu, Chunhong Zhang, and Benhui Zhuang. 2023. Hierarchical Reinforcement Learning With Attention Reward: Extended abstract. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 3 pages.

1 INTRODUCTION

Hierarchical reinforcement learning (HRL) plays an important role in solving complex tasks by breaking a difficult task into simple subtasks. As one of the successful HRL paradigms, subgoal-based HRL provides low-level agents a subgoal in the state space, which is the action of the high-level agent. Although these approaches are capable of learning demanding tasks with significant efficiency, training hierarchical agents still requires countless amounts of interaction, making sample efficiency a vital concern in HRL.Conventional subgoal-based HRL approaches generally evaluate the intrinsic reward based on the distance between states and subgoals [4, 5, 8, 10]. However, none of them consider task-specified knowledge in their intrinsic reward, which limits their sample efficiency.

In this paper, we propose a novel HRL method called Hierarchical reinforcement learning with Attention Reward (HiAR) to tackle this problem. The idea of attention reward originates from the notion of *contingency awareness* [1, 2, 9], the recognition that some aspects of the state can be affected by one's action. Since contingency awareness plays an important role in intelligence development, we believe the knowledge that concerns which part of the environment is affected by the action are vital for agent learning, and thus requires particular attention. Our attention reward follows this intuition, which aims to motivate agents to focus on the difference of states relevant to their control. We describe how we evaluate the attention reward based on task-relevant knowledge and integrate it into HRL to improve sample efficiency.

2 HIERARCHICAL REINFORCEMENT LEARNING WITH ATTENTION REWARD

In this section, we introduce three key elements in HiAR: the inverse dynamics model, attention reward and hindsight experience replay.

2.1 Inverse dynamics model

To distinguish how much a dimension in state space is affected by the action, we design an inverse dynamic model denoted M trained along with the low-level agent. The input of the model is states and subgoals in two continuous steps: s_t, s_{t+1} and g_t, g_{t+1}. The output of the model is the predicted action between the two steps $\hat{a_t}$. Each step the lower level agent stores a transition $(s_t, s_{t+1}, a_t, g_t, g_{t+1}, r^l_t)$ for off-policy training, both the input of the inverse dynamic model

[*] Corresponding author

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom. © 2023 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

and the ground-truth action is the action a_t stored in the transition, so we can the inverse dynamics model along with the low-level agent. The inverse dynamics model can be self-supervised and optimized with the mean squared error loss without extra data or supervision labels.

2.2 Attention Reward

We use the predicted action $\hat{a}_t$ to generate contingent weight which represents how much each dimension is affected by the action. We adapt the representation erasure method [6] to analyze the contingent weight based on the predict action $a_{t,\hat{\neg}k}$:

$$a_{t,\hat{\neg}k} = M(s_{t,\neg k}, s_{t+1,\neg k}, g_{t,\neg k}, g_{t+1,\neg k}) \tag{1}$$

Where $s_{t,\neg k}$ is the input vector with k-th element erased, that is, set to zero. By successively erasing each dimension in input vectors, we can perform error analysis on these predicted actions. Comparing the predicted action before and after a dimension gets erased, we evaluate how important a dimension is when predicting the primitive action a_t. For each predicted action $a_{t,\hat{\neg}k}$, we calculate its likelihood to the ground-truth action a_t. The likelihood L is defined as follows:

$$L(\hat{a}_t, a_t) = -\|\hat{a}_t - a_t\|_2 \tag{2}$$

The importance of dimension k denoted by $I_t(k)$ is evaluated based on the difference between $L(\hat{a}_t, a_t)$ and $L(\hat{a}_{t,\neg k}, a)$:

$$I_t(k) = \frac{L(\hat{a}_t, a_t) - L(\hat{a}_{t,\neg k}, a_t)}{L(\hat{a}_t, a_t)} \tag{3}$$

Now we have the importance of all dimensions, they will be converted into contingent weight W_t using softmax operator: $W_t = softmax(I_t)$. In HiAR, the low-level agent is rewarded for getting close to the subgoal with a dense intrinsic reward $r_{i,t} = -\|s_t + g_t - s_{t+1}\|_2$. To motivate our agent to pay more attention to the dimensions affected by its actions, we generate an attention reward for the low-level agent based on the contingent weight. The attention reward $r_{a,t}$ for the low-level agent is defined as:

$$r_{a,t} = -\|s_t + g_t - s_{t+1}\| \cdot W_t \tag{4}$$

We adjust the ratio between intrinsic reward and attention reward with β_1, β_2, resulting in the shaped reward r_t^l for the low-level agent: $r_t^l = \beta_1 \cdot r_{i,t} + \beta_2 \cdot r_{a,t}$. We suggest setting β_1 to 2 and β_2 to 1, which generally achieves good performances.

2.3 Hindsight experience replay

To improve the performance of HiAR in sparse reward environments, we perform hindsight action transition [5] for our high-level agent. However, our subgoal is the relative distance between the current state and desired state, and our reward is not binary, so we set the hindsight action $\bar{a} = s_{t+c} - s_t$ for the higher level agent, c denotes the frequency for the high-level agent to take action.

Considering the sparse reward challenge, we also replace the goal of the high-level agent with the state reached at the end of the subgoal to get an extrinsic reward different from -1, pretending the overall goal has been achieved. So, the hindsight transition of our high-level agent is:

$$(s_t, a_t, s_{t+c}, g_t, r_t) \Rightarrow (s_t, \bar{a}_t, s_{t+c}, \bar{g}_t := s_{t+c}, \bar{r}_t) \tag{5}$$

Here, $\bar{r}_t$ is the received extrinsic reward which supposed the overall goal is s_{t+c} and has been reached. For each c step, we add this extra transition to the replay buffer of the high-level agent.

3 EXPERIMENT

We evaluate HiAR in UR5 Reacher, Pendulum tasks in MuJoCo [3].

3.1 Contingent weight visualization

We visualize the contingent weight on each time step during a specified subtask in UR5 Reacher. The action is the force applied on each joint. We let a well-trained agent reach a specified subgoal, the initial state is manually set to make the subgoals reachable by rotating a single joint. The heatmaps of contingent weight are shown in Figure1, where the rotated joint always receives the highest contingent weight when the agent controls it to finish the subtask, confirming our contingent weight is interpretable.

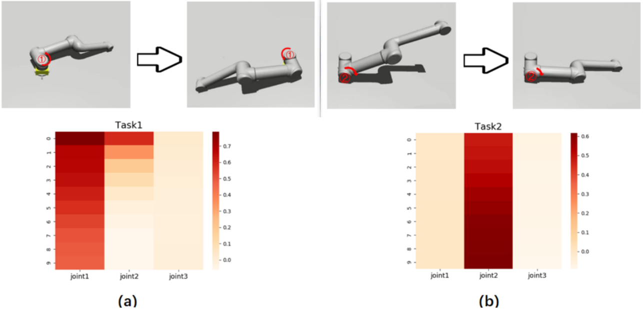

Figure 1: The heatmap of contingent weight on the angular velocity of each joint in two subtasks of UR5 Reacher

3.2 Comparative Analysis

In this section, we test our HiAR against HAC [5] and HIRO [8] in the Pendulum, UR5 Reacher environments, all HRL algorithms are implemented in two-level hierarchy with DDPG [7] and using the same hyper-parameters. the result is shown in Figure2. In both tasks, the sample efficiency of HiAR significantly outperforms baseline HRL methods.

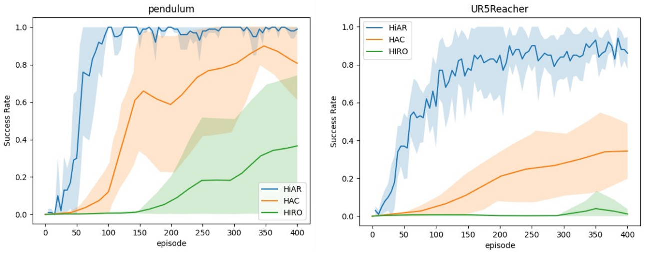

Figure 2: Performance of HiAR and baseline HRL methods

ACKNOWLEDGMENTS

This work was supported by the Guangdong Province Science and Technology Project 2021A0505080015.

REFERENCES

[1] Frank Baeyens, Paul Eelen, and Omer van den Bergh. 1990. Contingency awareness in evaluative conditioning: A case for unaware affective-evaluative learning. *Cognition and emotion* 4, 1 (1990), 3–18.

[2] Marc G Bellemare, Joel Veness, and Michael Bowling. 2012. Investigating contingency awareness using Atari 2600 games. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

[3] Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. 2016. Benchmarking deep reinforcement learning for continuous control. In *International conference on machine learning*. PMLR, 1329–1338.

[4] Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. 2016. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. *Advances in neural information processing systems* 29 (2016).

[5] Andrew Levy, George Konidaris, Robert Platt, and Kate Saenko. 2019. Learning multi-level hierarchies with hindsight. In *Proceedings of International Conference on Learning Representations*.

[6] Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220* (2016).

[7] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015).

[8] Ofir Nachum, Shixiang Shane Gu, Honglak Lee, and Sergey Levine. 2018. Data-efficient hierarchical reinforcement learning. *Advances in neural information processing systems* 31 (2018).

[9] John S Watson. 1966. The development and generalization of" contingency awareness" in early infancy: Some hypotheses. *Merrill-Palmer Quarterly of Behavior and Development* 12, 2 (1966), 123–135.

[10] Tianren Zhang, Shangqi Guo, Tian Tan, Xiaolin Hu, and Feng Chen. 2020. Generating adjacency-constrained subgoals in hierarchical reinforcement learning. *Advances in Neural Information Processing Systems* 33 (2020), 21579–21590.