Efficient Deep Reinforcement Learning through Policy Transfer

Extended Abstract

Tianpei Yang, Jianye Hao*,
Zhaopeng Meng
College of Intelligence and
Computing, Tianjin University
{tpyang,jianye.hao,mengzp}@tju.edu.cn

Zongzhang Zhang
Nanjing University
zzzhang@nju.edu.cn

Yujing Hu, Yingfeng Chen,
Changjie Fan
Fuxi AI Lab in Netease
{huyujing,chenyingfeng1,
fanchangjie}@corp.netease.com

Weixun Wang
Tianjin University
wxwang@tju.edu.cn

Zhaodong Wang
Washington State University
zhaodong.wang@wsu.edu

Jiajie Peng
Northwestern Polytechnical
University
jiajiepeng@nwpu.edu.cn

ABSTRACT

Transfer Learning (TL) has shown great potential to accelerate Reinforcement Learning (RL) by leveraging prior knowledge from past learned policies of relevant tasks. Existing TL approaches either explicitly computes the similarity between tasks or select appropriate source policies to provide guided explorations for the target task. However, how to directly optimize the target policy by alternatively utilizing knowledge from appropriate source policies without explicitly measuring the similarity is currently missing. In this paper, we propose a novel Policy Transfer Framework (PTF) by taking advantage of this idea. PTF learns when and which source policy is the best to reuse for the target policy and when to terminate it by modeling multi-policy transfer as the option learning problem. PTF can be easily combined with existing deep RL approaches. Experimental results show it significantly accelerates the learning process and outperforms state-of-the-art policy transfer methods in both discrete and continuous action spaces.

KEYWORDS

Reinforcement learning; Policy transfer; Policy reuse

ACM Reference Format:
Tianpei Yang, Jianye Hao*, Zhaopeng Meng, Zongzhang Zhang, Yujing Hu, Yingfeng Chen, Changjie Fan, Weixun Wang, Zhaodong Wang, and Jiajie Peng. 2020. Efficient Deep Reinforcement Learning through Policy Transfer. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 3 pages.

1 INTRODUCTION

Recent advance in Deep Reinforcement Learning (DRL) has obtained expressive success of achieving human-level control in complex tasks [7, 9]. However, DRL is still faced with sample inefficiency problems especially when the state-action space becomes large, which makes it difficult to learn from scratch. TL has shown great potential to accelerate RL [13] via leveraging prior knowledge from past learned policies of relevant tasks [4, 10, 16]. One major direction of transfer in RL focused on measuring the similarity between two tasks either through mapping the state spaces between two tasks [2, 17], or computing the similarity of two Markov Decision Processes (MDPs) [12], and then transferring value functions directly according to their similarities.

Another direction of policy transfer focuses on selecting a suitable source policy for explorations [3, 6]. However, such single-policy transfer cannot be applied to cases when one source policy is only partially useful for learning the target task. Although some transfer approaches utilized multiple source policies during the target task learning, they suffer from some limitations, e.g., Laroche and Barlier [4] assumed that all tasks share the same transition dynamics and differ only in the reward function; Li et al. [5] proposed Context-Aware Policy reuSe (CAPS) which required the optimality of source policies. Furthermore, it manually adds primitive policies to the policy library which limits its generality and cannot be applied to problems of continuous action spaces.

To address the above problems, we propose a novel Policy Transfer Framework (PTF) which combines the above two directions of policy reuse. Instead of using source policies as guided explorations in a target task, we adaptively select a suitable source policy during target task learning and use it as a complementary optimization objective of the target policy. The backbone of PTF can still use existing DRL algorithms to update its policy, and the source policy selection problem is modeled as the option learning problem. In this way, PTF does not require any source policy to be perfect on any subtask and can still learn toward an optimal policy in case none of the source policy is useful. Besides, we propose an adaptive and heuristic mechanism to ensure the efficient reuse of source policies and avoid negative transfer.

* Corresponding author, secondary affiliation: Noah's Ark Lab, Huawei.

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand.

2 POLICY TRANSFER FRAMEWORK

We describe PTF applying in A3C [8]: PTF-A3C in detail. Our proposed Policy Transfer Framework (PTF) contains two main components, one is the agent module (here is an example of an A3C model), which is used to learn the target policy with guidance from the option module. The other is the option module, which is used to learn when and which source policy is useful for the agent module. Given a set of source policies $\Pi_s = \{\pi_1, \pi_2, \cdots, \pi_n\}$ as the intra-option policies, the PTF-A3C agent first initializes a set of options $\mathcal{O} = \{o_1, o_2, \cdots, o_n\}$ together with the option-value network with random parameters. At each step, it selects an action following its policy, and also selects an option o_i according to the option-value function and the termination probabilities. For the update, except for calculating the original A3C loss, the PTF-A3C agent introduces a complementary loss $\mathrm{L_H}$ which transfers knowledge from the intra-option policy π_i through imitation, weighted by an adaptive adjustment factor. The reuse of the policy π_i terminates according to the termination probability of o_i and then another option is selected for reuse following the policy over options, which is ϵ-greedy to the option-value Q_o.

The remaining issue is how to update the option-value network and the termination network. First, the agent samples a batch of N transitions from the replay buffer and updates the option-value network by minimizing the loss: $L = \frac{1}{N}\sum_i(r + \gamma U(s', o|\theta_o) - Q_o(s_i, o|\theta_o))^2$, where $U(s', o|\theta_o) = (1 - \beta(s', o|\theta_\beta))Q'_o(s', o|\theta'_o) + \beta(s', o|\theta_\beta)\max_{o'\in O} Q'_o(s', o'|\theta'_o)$ [14]. The objective of learning the termination probability is to maximize the expected return U, so the termination network parameters θ_β is updated as follows: $\theta_\beta = \theta_\beta - \alpha_\beta \frac{\partial \beta(s', o|\theta_\beta)}{\partial \theta_\beta}\left(A(s', o|\theta_o) + \xi\right)$, where α_β is the learning rate, $A(s', o|\theta_o)$ is the advantage function, ξ is a regularization term which is used to ensure sufficient exploration that the best option could be selected.

Finally, we propose an adaptive and heuristic way to transfer knowledge from the selected source policy. Specifically, we propose adaptively adjust the weighting factor $f(\beta_o, t)$ of the complementary loss L_H as follows: $f(\beta_o, t) = f(t)(1 - \beta(s_t, o|\theta_\beta))$, where $f(t)$ is a discount function. When the value of the termination function of option o increases, it means that the performance of the option o is not the best one among all options based on the current experience. Thus we decrease the weighting factor $f(\beta_o, t)$ and vice versa. $f(t)$ controls the slow decrease in exploiting the transferred knowledge from source policies which means at the beginning of learning, we exploit source knowledge mostly. As learning continues, past knowledge becomes less useful and we focus more on the current self-learned policy. In this way, PTF efficiently exploits useful information and avoids negative transfer from source policies.

3 EXPERIMENTAL RESULTS

In this section, we evaluate PTF on three test domains, grid world [3], pinball [1] and reacher [15] compared with several DRL methods learning from scratch (A3C [8] and PPO [11]); and the state-of-the-art policy transfer method CAPS [5], implemented as a deep version (Deep-CAPS). Results are averaged over 20 random seeds [1].

Table 1: Average rewards with std.dev.(±) in three domains.

Games / Methods	Grid world	Pinball	Reacher
A3C	3.9±0.03	2.4±0.35	50.9±7.3
PTF-A3C	**4.0±0.02**	**6.7±0.05**	**59.5±3.2**
Deep CAPS	3.9±0.54	4.7±0.36	
PPO	3.9±0.02	2.2±0.24	51.6±8.7
PTF-PPO	**4.0±0.01**	**6.6±0.03**	**61.4±3.9**

Table 1 presents the average discounted rewards of different algorithms on three test domains. We can see that PTF-A3C significantly accelerates the learning process and outperforms A3C and CAPS. Similar results can be found that PTF-PPO outperforms PPO. The reason is that PTF enables the agent to quickly identify the optimal source policy and exploit useful information from source policies, which efficiently accelerates the learning process than learning from scratch. The performance gap between PTF-A3C and deep-CAPS is because the policy reuse module and the target task learning module in PTF are loosely decoupled, apart from reusing knowledge from source policies, PTF is also able to utilize its own experience from the environment. However, in deep-CAPS, these two parts are highly decoupled, which means its explorations and exploitations are fully dependent on the source policies inside the options. Thus, deep-CAPS needs higher requirements on source policies than our PTF, and finally achieves lower performance than PTF-A3C.

4 CONCLUSION AND FUTURE WORK

In this paper, we propose a Policy Transfer Framework (PTF) which can efficiently select the optimal source policy and exploit the useful information to facilitate the target task learning. PTF also efficiently avoids negative transfer through terminating the exploitation of current source policy and selects another one adaptively. PTF can be easily combined with existing deep policy-based and actor-critic methods. Experimental results show PTF efficiently accelerates the learning process of existing state-of-the-art DRL methods and outperforms previous policy reuse approaches. As a future topic, it is worthwhile investigating how to extend PTF to multiagent settings. Another interesting direction is how to learn abstract knowledge for fast adaptation in new environments.

ACKNOWLEDGMENTS

The work is supported by the National Natural Science Foundation of China (Grant Nos.: 61702362, U1836214).

[1] The extended version is put on http://arxiv.org/abs/2002.08037

REFERENCES

[1] Pierre-Luc Bacon, Jean Harb, and Doina Precup. 2017. The Option-Critic Architecture. In *Proceedings of AAAI Conference on Artificial Intelligence*. 1726–1734.

[2] Tim Brys, Anna Harutyunyan, Matthew E Taylor, and Ann Nowé. 2015. Policy transfer using reward shaping. In *Proceedings of International Conference on Autonomous Agents and Multiagent Systems*. 181–188.

[3] Fernando Fernández and Manuela Veloso. 2006. Probabilistic Policy Reuse in a Reinforcement Learning Agent. In *Proceedings of International Conference on Autonomous Agents and Multiagent Systems*. 720–727.

[4] Romain Laroche and Merwan Barlier. 2017. Transfer Reinforcement Learning with Shared Dynamics. In *Proceedings of AAAI Conference on Artificial Intelligence*. 2147–2153.

[5] Siyuan Li, Fangda Gu, Guangxiang Zhu, and Chongjie Zhang. 2019. Context-Aware Policy Reuse. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems*. 989–997.

[6] Siyuan Li and Chongjie Zhang. 2018. An Optimal Online Method of Selecting Source Policies for Reinforcement Learning. In *Proceedings of AAAI Conference on Artificial Intelligence*. 3562–3570.

[7] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2016. Continuous control with deep reinforcement learning. In *Proceedings of International Conference on Learning Representations*.

[8] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *Proceedings of International Conference on Machine Learning*. 1928–1937.

[9] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529.

[10] Janarthanan Rajendran, Aravind S Lakshminarayanan, Mitesh M Khapra, P Prasanna, and Balaraman Ravindran. 2017. Attend, Adapt and Transfer: Attentive Deep Architecture for Adaptive Transfer from multiple sources in the same domain. In *Proceedings of International Conference on Learning Representations*.

[11] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[12] Jinhua Song, Yang Gao, Hao Wang, and Bo An. 2016. Measuring the distance between finite Markov decision processes. In *Proceedings of International Conference on Autonomous Agents and Multiagent Systems*. 468–476.

[13] Richard S Sutton and Andrew G Barto. 1998. *Reinforcement learning: An introduction*. MIT press.

[14] Richard S. Sutton, Doina Precup, and Satinder Singh. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence* 112, 1 (1999), 181 – 211.

[15] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy P. Lillicrap, and Martin A. Riedmiller. 2018. DeepMind Control Suite. *CoRR* abs/1801.00690 (2018).

[16] Matthew E Taylor and Peter Stone. 2009. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research* 10, Jul (2009), 1633–1685.

[17] Matthew E Taylor, Peter Stone, and Yaxin Liu. 2007. Transfer learning via inter-task mappings for temporal difference learning. *Journal of Machine Learning Research* 8, Sep (2007), 2125–2167.