Online Learning of Shaping Reward with Subgoal Knowledge

Extended Abstract

Takato Okudo
The Graduate University for Advanced Studies, SOKENDAI
Tokyo, Japan
okudo@nii.ac.jp

Seiji Yamada
National Institute of Informatics, NII
The Graduate University for Advanced Studies, SOKENDAI
Tokyo, Japan
seiji@nii.ac.jp

ABSTRACT

SARSA-RS is a reward shaping method that updates the shaping through learning. However, the bottleneck of this method is the aggregation of states since designers need to design mappings from all states to abstract states. We propose a dynamic trajectory aggregation that uses subgoal series. The designer's effort becomes minimal because only human input is the subgoal series. This makes application to environments with high-dimensional observations possible. We compared our method by using participants' subgoal series with a baseline reinforcement learning algorithm and other subgoal-based methods in a navigation task. As a result, our reward shaping outperformed all other methods in learning efficiency.

KEYWORDS

Reinforcement Learning, Reward Shaping, Subgoal

ACM Reference Format:
Takato Okudo and Seiji Yamada. 2021. Online Learning of Shaping Reward with Subgoal Knowledge: Extended Abstract. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021*, IFAAMAS, 3 pages.

1 INTRODUCTION

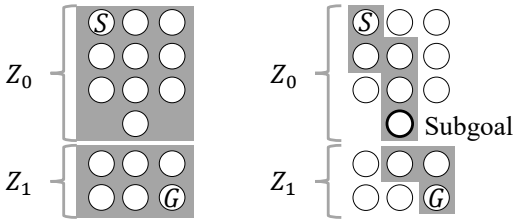
Reinforcement learning(RL) can acquire a policy maximizing long-term rewards. It is common for the state-action space to be quite large in a real environment. For that case, the learning becomes too slow to obtain optimal policies in a realistic amount of time. Since a human could have knowledge helpful to an agent in such cases, a promising approach is utilizing human knowledge [11, 15, 20].

The reward function is most related to learning efficiency. Most difficult tasks in RL have a sparse reward function [3]. Inverse reinforcement learning (IRL) [1, 18] is the most popular method for enriching the reward function. IRL uses an optimal policy to generate a dense reward function [13, 21]. There is the question of the teacher's cost in providing trajectories or policies. Humans sometimes have difficulty providing these because they may not have the skills. In particular, in a robotics task, humans are required to have robot-handling skills and knowledge on the optimal trajectory. Another approach is reward shaping [6–8, 12, 16]. Potential-based reward shaping(PBRS) is able to add external rewards while keeping the optimal policy of the environment [17]. To use PBRS, we need to define the potential function. SARSA-RS acquires it in learning [9, 10]. SARSA-RS needs a function for aggregating states. However, this is often unavailable when the task has a high-dimensional observation. We propose a subgoal-based trajectory aggregation method. Our method needs only ordered states as subgoal series. The way of providing the external knowledge with our method is easier than state aggregation because access to all states is not required. Since humans only provide several states, skill in controlling robots are not always necessary. This may give non-expert a chance to enhance RL algorithms from viewpoint of Interactive RL [2].

2 SUBGOAL-BASED DYNAMIC TRAJECTORY AGGREGATION

We propose dynamic trajectory aggregation from states into abstract states with subgoal series. The method basically follows SARSA-RS [9], and the difference is mainly the aggregation function and minorly the accumulated rewards. Our method aggregates trajectories dynamically into abstract states during learning with subgoal series. We define a subgoal as a state s that is a *subgoal* if s is a goal in one of the subtasks decomposed from a task. A subgoal series is written formally as $(SG, \prec)$. SG is a set of subgoals and a sub-set of S. There are two types of subgoal series, totally ordered and partially ordered. With totally ordered subgoals, a subgoal series is deterministically determined at any subgoal. In contrast, partially ordered subgoals have several transitions to the subgoal series from a subgoal. We use only the totally ordered subgoal series, but both types of ordered subgoals can be used for our method. Since an agent achieves a subgoal only once, the transition between subgoals is unidirectional. We assume that the subgoal series $(SG, \prec)$ is pre-defined, and $(SG, \prec) = \{sg_0 \prec sg_1 \prec \cdots \prec sg_n\}$.

Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, Online.



(a) Aggregation in SARSA-RS (b) Our Aggregation

Figure 1: Concept of subgoal-based aggregation.

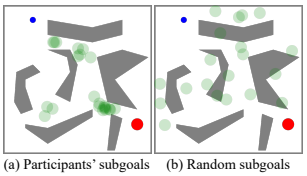


Figure 2: Subgoal distribution in pinball domain.

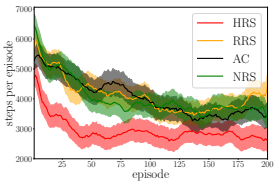


Figure 3: Learning curves in pinball domain. Line is mean, and shaded area is standard error

2.1 Dynamic Trajectory Aggregation

We build abstract states to represent the achievement status of a subgoal series. If there are n subgoals, the size of abstract states is $n+1$. The agent is in a first abstract state z_0 before it achieves a subgoal. Then, the abstract state z_0 transits to z_1 when the subgoal sg_0 is achieved. This means that trajectories are aggregated until subgoal sg_0 transits into z_0. The aggregated trajectories change dynamically every trial because of the policy with randomness. As the learning progresses, the aggregated trajectories become fixed. The value over abstract states is distributed to the values of states of the trajectory. Note that the trajectories for updating the value are different from those that the values are distributed to. The updated value function is not used for the current trial but for the next trials. An image of the dynamic trajectory aggregation is shown in Figure1. As is shown, a circle is a state, and the aggregated states are in each gray background area. The bold circles express the states with which the designer deals. The number of bold circles in Figure 1(b) is much lower than Figure 1(a). "S" and "G" in the circles are a start and a goal, respectively. Figure 1(b) shows that the trajectory is separated into two sub-episodes, and each of them corresponds to abstract states.

2.2 Accumulated Reward Function

We clearly define the reward transformation function r_h because our method only updates the achievements of subgoals. We describe this formally as $r_h = \sum_{t=0}^{N-1} \gamma^t r_t$, where N is the duration until subgoal achievement. The function accumulates rewards with discount γ. Depending on the policy at the time, N is varied dynamically. This follows n-step temporal difference (TD) learning [19] because there are transitions between an abstract state z_i and another one z_{i+1}.

3 EXPERIMENTS

We used a navigation task in pinball in OpenAI Gym [5]. Details can be found in [4]. We used a reward function generating a reward of +10000 when the agent reached the goal. This domain is difficult for humans because delicate control of the ball is necessary. We acquired ten patterns of two ordered subgoals from participants, and the evaluation was conducted using them with a PC [Core i7-7700 (3.6GHz), 16 GB of memory]. We compared our method with human subgoals (HRS) with three other methods, an actor-critic algorithm (AC) [14], our method with random subgoals (RRS), and naive subgoal reward shaping (NRS). The three reward shaping methods were implemented with AC. RRS used ten patterns of two randomly selected subgoals from the whole state space. NRS has the potential function $\Phi(s)$ that outputs a scalar value η just when an agent has visited a subgoal state, and 0 otherwise. We set η as 10,000 so as to be the same value as the goal reward. The difference from our method is that there was a fixed value only for subgoals. All methods learned a total 100 times from scratch through 200 episodes. The learning took several tens of minutes. A subgoal had only a center position and a radius that was the same as the target. The agent achieved a subgoal when it entered the circle of the subgoal at any velocity. The setting of AC was the same as [4].

Figure 2 shows the subgoal distribution acquired from ten participants and from the random subgoals generated. In this figure, the color of the start point, the goal, and the subgoals are red, blue, and green, respectively. As shown in the figure, participants focused on four regions of branch points to set subgoals in comparison with random subgoals.

Figure 3 shows the learning curves of HRS, RRS, AC, and NRS. The learning indicator was the average number of steps per episode over learning 100 times. It took an average shift of 10 episodes. As is shown, HRS performed better than all other methods. RRS and NRS seemed to be almost the same as AC.

4 CONCLUSION

Although SARSA-RS incorporating state aggregation information into rewards is helpful, humans will rarely have to deal with all states in an environment with larger continuous observations. We proposed dynamic trajectory aggregation by which a human deals with several characteristic states as a subgoal. We evaluated a navigation task involving pinball with subgoals from participants. The experimental results revealed that our method with human subgoals enabled faster learning compared with the others. Our method could make SARSA-RS available without the mapping of all states, and learning was clearly accelerated. We plan to apply our method to an environment with image observations.

REFERENCES

[1] Pieter Abbeel and Andrew Y. Ng. 2004. Apprenticeship Learning via Inverse Reinforcement Learning. In *Proceedings of the 14th International Conference on Machine Learning*. Association for Computing Machinery, 1.

[2] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. 2014. Power to the People: The Role of Humans in Interactive Machine Learning. (2014), 105–120.

[3] Yusuf Aytar, Tobias Pfaff, David Budden, Tom Le Paine, Ziyu Wang, and Nando de Freitas. 2018. Playing Hard Exploration Games by Watching YouTube. In *Advances in Neural Information Processing Systems 31*. Curran Associates Inc., 2935âĂŞ2945.

[4] Pierre-Luc Bacon, Jean Harb, and Doina Precup. 2017. The Option-critic Architecture. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*. 1726–1734.

[5] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. OpenAI Gym. arXiv:arXiv:1606.01540

[6] Alper Demir, Erkin Çilden, and Faruk Polat. 2019. Landmark Based Reward Shaping in Reinforcement Learning with Hidden States. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. 1922–1924.

[7] Sam Devlin and Daniel Kudenko. 2012. Dynamic Potential-based Reward Shaping. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*. 433–440.

[8] Yang Gao and Francesca Toni. 2015. Potential Based Reward Shaping for Hierarchical Reinforcement Learning. *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, 3504–3510.

[9] Marek Grzes and Daniel Kudenko. 2010. Online learning of shaping rewards in reinforcement learning. *Neural networks* 23 (2010), 541–550.

[10] Marek GrzeÅŻ and Daniel Kudenko. 2008. Multigrid Reinforcement Learning with Reward Shaping. *Lecture Notes in Computer Science* (2008), 357–366.

[11] Anna Harutyunyan, Tim Brys, Peter Vrancx, and Ann Nowé. 2015. Shaping Mario with Human Advice. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, Istanbul, Turkey, May 4-8, 2015*. ACM, 1913–1914.

[12] Anna Harutyunyan, Sam Devlin, Peter Vrancx, and Ann Nowe. 2015. Expressing Arbitrary Reward Functions as Potential-Based Advice. *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, 2652–2658.

[13] Jonathan Ho and Stefano Ermon. 2016. Generative Adversarial Imitation Learning. In *Advances in Neural Information Processing Systems 29*. Curran Associates, Inc., 4565–4573.

[14] Vijay R. Konda and John N. Tsitsiklis. 2000. Actor-Critic Algorithms. *Advances in Neural Information Processing Systems 12* (2000), 1008–1014.

[15] G. Li, R. Gomez, K. Nakamura, and B. He. 2019. Human-Centered Reinforcement Learning: A Survey. *IEEE Transactions on Human-Machine Systems* 49, 4 (2019), 337–349.

[16] Siyuan Li, Rui Wang, Minxue Tang, and Chongjie Zhang. 2019. Hierarchical Reinforcement Learning with Advantage-Based Auxiliary Rewards. *Advances in Neural Information Processing Systems 32* 32 (2019), 1409–1419.

[17] Andrew Y. Ng, Daishi Harada, and Stuart J. Russell. 1999. Policy Invariance Under Reward Transformations: Theory and Application to Reward Shaping. In *Proceedings of the 16th International Conference on Machine Learning*. 278–287.

[18] Andrew Y. Ng and Stuart J. Russell. 2000. Algorithms for Inverse Reinforcement Learning. In *Proceedings of the 17th International Conference on Machine Learning*. 663–670.

[19] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction* (second ed.). The MIT Press.

[20] Matthew E. Taylor. 2018. Improving Reinforcement Learning with Human Input. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 5724–5728.

[21] Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind K. Dey. 2008. Maximum Entropy Inverse Reinforcement Learning. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*. 1433–1438.