Hierarchical Reinforcement Learning with Integrated Discovery of Salient Subgoals

Extended Abstract

Shubham Pateria
Nanyang Technological University
Singapore
SHUBHAM007@e.ntu.edu.sg

Budhitama Subagdja
Nanyang Technological University
Singapore
budhitama@ntu.edu.sg

Ah-Hwee Tan
Nanyang Technological University
Singapore
asahtan@ntu.edu.sg

ABSTRACT

Hierarchical Reinforcement Learning (HRL) is a promising approach to solve more complex tasks which may be challenging for the traditional reinforcement learning. HRL achieves this by decomposing a task into shorter-horizon subgoals which are simpler to achieve. Autonomous discovery of such subgoals is an important part of HRL. Recently, end-to-end HRL methods have been used to reduce the overhead from offline subgoal discovery by seeking the useful subgoals while simultaneously learning optimal policies in a hierarchy. However, these methods may still suffer from slow learning when the search space used by a high level policy to find the subgoals is large. We propose LIDOSS, an end-to-end HRL method with an integrated heuristic for subgoal discovery. In LIDOSS, the search space of a high level policy can be reduced by focusing only on the subgoal states that have high saliency. We evaluate LIDOSS on continuous control tasks in the MuJoCo Ant domain. The results show that LIDOSS outperforms Hierarchical Actor Critic (HAC), a state-of-the-art HRL method, in the fixed goal tasks.

KEYWORDS

Hierarchical Reinforcement Learning; Reinforcement Learning; Subgoal discovery

ACM Reference Format:
Shubham Pateria, Budhitama Subagdja, and Ah-Hwee Tan. 2020. Hierarchical Reinforcement Learning with Integrated Discovery of Salient Subgoals. 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

Hierarchical Reinforcement Learning (HRL) is a promising approach to learn behaviour policies in the long-horizon or sparse reward tasks, by decomposing the task goals into simpler subgoals through a hierarchy of policies. A challenging aspect of HRL is the specification of the candidate subgoals. Traditional approaches [1, 2, 12] relied on hand-crafted subgoals, which reduce the autonomy of a HRL method. Subsequent research addresses this limitation through *subgoal discovery* [3, 6–8, 10, 11] in which the candidate subgoals for task decomposition are autonomously extracted through the interaction of the agent with its environment.

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.

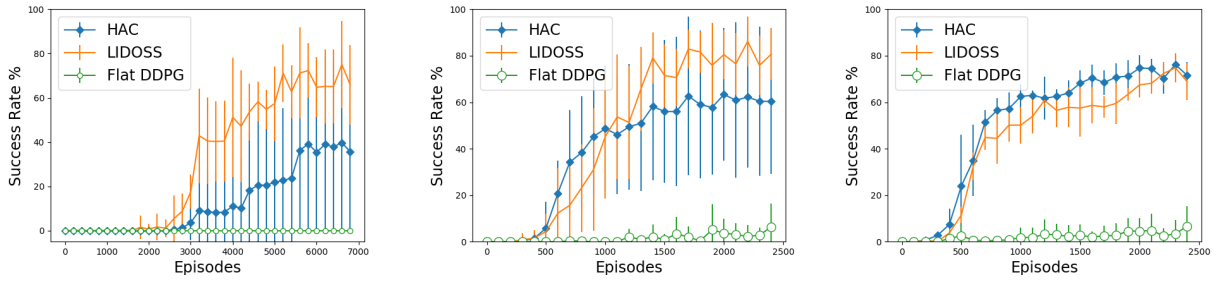
Most of the existing subgoal discovery approaches, however, separate the discovery process from the learning of the policy hierarchy. This may require more training time and computation due to the decoupled stages of discovery and learning. On the other hand, recent end-to-end HRL approaches eschew subgoal discovery in favour of training a policy hierarchy end-to-end [4, 9, 14]. These methods do not require explicit subgoals discovery. Instead, an agent uses a *large* subgoal space as the output space of a higher level policy. As this policy is optimized over time, it learns to select useful subgoals to achieve a given goal. While such approaches reduce the overhead of a decoupled subgoal discovery stage, the convergence of a higher level policy may still be slow due to the use of a large subgoal space.

We propose a method for *in situ* subgoal discovery with the end-to-end learning of a policy hierarchy in episodic goal-directed tasks. This is achieved by calculating the frequency of occurrence of an intermediate state on the trajectories leading to a goal state. The states with relatively high frequencies are extracted as *salient subgoals*, with the intuition that reaching these states might improve the rate of achieving the goal in the future. The subgoal discovery is performed incrementally using the experience trajectories gathered while the agent learns. Firstly, this approach provides an iterative integration of subgoal discovery process with the hierarchical learning and circumvents the need of decoupled stages. Secondly, the highest level policy can guide the behaviour along the salient subgoals aligned with the goal-reaching trajectories, rather than searching subgoals in a large state space.

The proposed method is named Hierarchical Reinforcement **L**earning with **I**ntegrated **D**iscovery **O**f **S**alient **S**ubgoals (LIDOSS).

2 METHOD

The proposed LIDOSS agent consists of a multi-level policy hierarchy and a Subgoal Discovery Module (SDM) working in an integrated manner (Figure 2). The policy hierarchy generates the behaviour of the agent, which is observed by the SDM as episodic trajectories. The SDM uses this as experience data to extract salient subgoals into a set g_{SDM} which is used by the highest level policy as the output space for the further episodes of learning. This results in new experience data which is again used by SDM to refine or expand the subgoal set g_{SDM}. This iterative process results in an *in situ* subgoal discovery in integration with the training of the policy hierarchy, rather than decoupled stages. This is essentially an end-to-end HRL approach, but unlike the existing standard end-to-end HRL methods [4, 9, 14], LIDOSS explicitly uses an integrated



(a) Ant Maze fixed goal. (b) Ant Four Rooms fixed goal. (c) Ant Four Rooms multi goal

Figure 1: Success rate in Ant Four Rooms and Ant Maze tasks. Each data point is an average of one test batch of 100 episodes. The results are averaged over ten randomly initiated trials. Flat DDPG is a non-hierarchical agent with only a primitive policy.

subgoal discovery heuristic to regulate the output space of the highest-level policy.

The subgoal discovery heuristic used in LIDOSS is based on the frequency/count of occurrence of a state on the successful trajectories which lead to the goal state G. Implicitly, a higher frequency implies a higher conditional probability $p(s|G)$ of observing a state s on the successful trajectories. To facilitate counting in a continuous state space $\mathbb{S}$, the state space is first discretized. Then, the discrete states with higher $p(s|G)$ are treated as salient subgoals. This requires ranking of the states using $p(s|G)$ values to find the salient ones. This ranking is done locally and a saliency value $\psi(s)$ is calculated using equation 1. Here, $LMX(s)$ is a Local Max kernel of size $|LMX|$, which when centered on a state s, consists of its $|LMX|$ neighbouring states. We use a spatial neighbourhood to constitute LMX. $\max_{LMX(s)} = \max_{s' \in LMX(s)} p(s'|G)$.

$$\psi(s) = \begin{cases} 0, & \text{if } p(s|G) < \max_{LMX(s)} \\ 0, & \text{if } \max_{LMX(s)} <= \\ & \quad p(s|G) < \textit{local threshold} \\ 1, & \text{otherwise} \end{cases} \tag{1}$$

We use $|LMX| = 24$ and $\textit{local threshold} = 0.3$. The SDM updates the salient subgoal set g_{SDM} periodically based on the ψ values, by adding the states with $\psi(s) = 1$ and removing those with saliency of zero. g_{SDM} is used as the output space of the highest level policy (as mentioned above).

3 EXPERIMENTS

We compare LIDOSS against a state-of-the-art HRL method Hierarchical Actor Critic (HAC) [4]. The comparison is done in terms of the success rate of achieving the goal. Three tasks, in the MuJoCo Ant continuous control domain [13], are used in the experiments. These are Ant Four rooms multi goal [4], Ant Four Rooms fixed goal, and Ant Maze. In the multi goal tasks, the goal state is changed in the beginning of each episode in a trial. In the fixed goal tasks, the goal is kept fixed for all episodes during a trial and changed only at the beginning of each trial. The results are shown in Figure 1.

We observe that the success rate of our LIDOSS agent increases faster than the baseline HAC agent in the first two tasks with

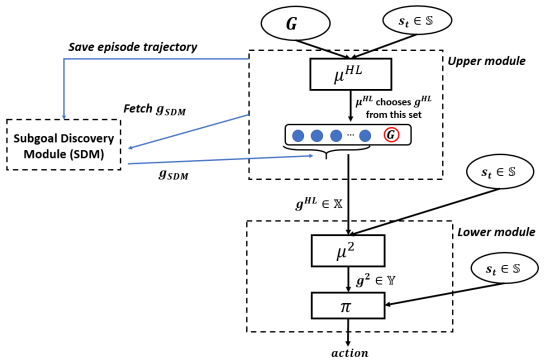


Figure 2: The structure of the LIDOSS agent. Subgoal discovery is performed at the highest level. The subgoal g^{HL} chosen by the highest level Deep Q-Network (DQN) μ^{HL} is taken as input by the intermediate-level DDPG [5] actor-critic μ^2 which generates a subgoal g^2 for the primitive DDPG actor-critic π. We follow the three-level structure similar to the baseline HAC three-level agent. $\mathbb{X}$ and $\mathbb{Y}$ are sub-spaces of $\mathbb{S}$.

fixed goals (Figure 1(a,b)). In these tasks, the space of the feasible successful trajectories in constrained due to both the topology of the environment (consisting of walls/obstacles) and the fixing of the goal state. Hence, an early discovery of salient subgoals (using the heuristic in LIDOSS) may result in faster convergence of the highest level policy μ^{HL} in contrast to the use of the entire state space as the output space of μ^{HL} (as in HAC). Whereas, the space of successful trajectories are less constrained in the Ant Four Rooms multi-goal task (Figure 1(c)) due to the goals being distributed over a larger space. Hence, useful subgoals, corresponding to various goals, effectively lie over the entire state space. This is possibly why the baseline method shows slightly better performance than LIDOSS because it learns a policy over the entire state space as subgoal space.

In conclusion, the proposed method, LIDOSS, with an integrated subgoal discovery heuristic may outperform a standard end-to-end HRL approach without explicit subgoal discovery when the task goals are fixed or lie in constrained regions of the state space.

REFERENCES

[1] Peter Dayan and Geoffrey E Hinton. 1993. Feudal reinforcement learning. In *Advances in neural information processing systems*. 271–278.

[2] Thomas G Dietterich. 2000. Hierarchical reinforcement learning with the MAXQ value function decomposition. *Journal of Artificial Intelligence Research* 13 (2000), 227–303.

[3] George Konidaris and Andrew G Barto. 2009. Skill discovery in continuous reinforcement learning domains using skill chaining. In *Advances in neural information processing systems*. 1015–1023.

[4] Andrew Levy, Robert Platt, and Kate Saenko. 2019. Hierarchical Reinforcement Learning with Hindsight. In *International Conference on Learning Representations*. https://openreview.net/forum?id=ryzECoAcY7

[5] 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).

[6] Marios C Machado, Marc G Bellemare, and Michael Bowling. 2017. A laplacian framework for option discovery in reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2295–2304.

[7] Shie Mannor, Ishai Menache, Amit Hoze, and Uri Klein. 2004. Dynamic abstraction in reinforcement learning via clustering. In *Proceedings of the twenty-first international conference on Machine learning*. ACM, 71.

[8] Amy McGovern and Andrew G Barto. 2001. Automatic Discovery of Subgoals in Reinforcement Learning using Diverse Density. In *Proceedings of the Eighteenth International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc., 361–368.

[9] Ofir Nachum, Shixiang Shane Gu, Honglak Lee, and Sergey Levine. 2018. Data-efficient hierarchical reinforcement learning. In *Advances in Neural Information Processing Systems*. 3303–3313.

[10] Scott Niekum, Sarah Osentoski, George Konidaris, Sachin Chitta, Bhaskara Marthi, and Andrew G Barto. 2015. Learning grounded finite-state representations from unstructured demonstrations. *The International Journal of Robotics Research* 34, 2 (2015), 131–157.

[11] Özgür Şimşek and Andrew G Barto. 2009. Skill characterization based on betweenness. In *Advances in neural information processing systems*. 1497–1504.

[12] 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, 1-2 (1999), 181–211.

[13] Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 5026–5033.

[14] Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. 2017. Feudal networks for hierarchical reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 3540–3549.