Transfer Learning Through Graph-based Skill Acquisition

Farzaneh Shoeleh
Faculty of Electrical and Computer Engineering
University of Tehran,Tehran, Iran
f.shoeleh@ut.ac.ir

Masoud Asadpour
Faculty of Electrical and Computer Engineering
University of Tehran,Tehran, Iran
asadpour@ut.ac.ir

ABSTRACT

Since Reinforcement Learning(RL) algorithms suffer from the curse of dimensionality in continuous domains, generalization is the most challenging issue in this area. Both skill acquisition and Transfer Learning(TL) are successful techniques to overcome such problem that result in big improvements in agent learning performance . In this paper, we propose a novel graph based skill acquisition method and a skill based transfer learning framework.

Keywords

Reinforcement Learning, Transfer Learning, Skill Acquisition

1. INTRODUCTION

The aim of this paper is to leverage TL techniques to make RL algorithms more efficient. As first step, the paper works towards a challenging question "How do we discover abstract skills automatically?". There are few works developed for continuous domain which an agent can discover useful new skills autonomously, and thereby construct its own high-level skill hierarchies [7, 6, 3, 4]. Here, we present a Graph based Skill Learning (GSL) method which allows agent to use graph representation in order to model its environment and find skills autonomously. Additionally we propose a transfer learning framework named Skill based Transfer Learning (STL) in continuous RL domain. In this framework, skills are evaluated based on their impact on leading the agent towards the goal state.

2. GRAPH BASED SKILL LEARNING

To extract high-level skills, a Graph based Skill Learning method (GSL) with three following phases is proposed.

Phase 1. Building Connectivity Graph. To construct Connectivity Graph (CG), two graph models, namely Transition Graph (TG) and Distance Graph (DG), are considered. The agent tries to build TG by gathering experiences via interaction with environment [5]. The problem domain properties would be captured through DG where the probability of existing a link between i and j, P_{ij} is defined as follows Formula 1. where d_{ij} is the Euclidean distance between nodes i and j. The CG is obtained as follows Formula 2, where $w_{i,j}$ is the weight of edge between node i and j.

$$P_{ij} = \frac{1}{1 + e^{d_{ij}}} \tag{1}$$

$$CG(w_{ij}) = \alpha_w * TG(w_{ij}) + (1 - \alpha_w) * DG(w_{ij}) \tag{2}$$

Phase 2. Identifying Skills based on Community Detection. If the communities of CG are found, each community present a region called *accessible regions* in the agent's state space, where states are accessible to each other through a limited number of actions. Here, each detected community is considered as a skill for the agent. A skill guides agent how to live in its *accessible regions* in order to enter the closest neighbor region to the goal region where contains the goal state. Therefore, agent faces with a new graph named *skill graph* where nodes corresponds to *accessible regions* and edges specify the relations between communities. Agent should learn how to optimally choose a sequence of skills to reach the goal, and learn how to act in each skill.

Phase 3. Skill Learning. After autonomously discovering new skills, the agent uses the option framework to learn each skill and thereby construct its own high level skill hierarchy. The option completion reward is considered as the value of the state where the option is terminated. This shaping reward encourages the agent to terminate the option and also leads it to a better neighbor option in skill graph which is closer to the goal.

3. SKILL BASED TRANSFER LEARNING

Skill based Transfer Learning (STL) approach contains three following phases.

Phase 1. Learning the Source Task. As the first step, agent learn the source task using GSL framework. Upon learning the source task, the agent is provided with a set of learned skills covering source problem state space.

Phase 2. Verifying the Source Skills Graph. To find which skill can be transferred into the target task, two methods are proposed to calculate the fitness of learned skills.

The first mechanism is Trajectory based Fitness (TF). It relies on how much a skill is successfully executed in target task. Agent collects some trajectories in form of $T = (s, o, r, s')$ from the source task and tries to apply them to the target environment. In the target domain, the agent verifies the collected tuples by executing action a in state s. If the agent reaches s' then this tuple is called *successful*

Appears in: *Proceedings of the 1st Workshop on Transfer in Reinforcement Learning (TiRL) at the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), A. Costa, D. Precup, M. Veloso, M. Taylor (chairs), May 8–9, 2017, São Paulo, Brazil.*

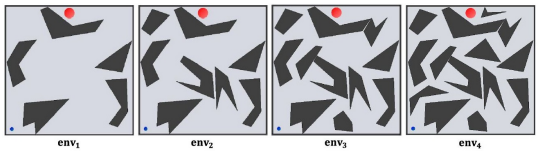


Figure 1: Pinball domain instances

tuple; otherwise it is an unsuccessful tuple. Having successful and unsuccessful tuples, the fitness of an option can be determined as follows:

$$Fitness(O) = \frac{\sum_{T is ussessful} h(T,O)}{\sum_T h(T,O)} \quad (3)$$

$$h(T,O) = \begin{cases} 1 & T.option = O \\ 0 & T.option \neq O \end{cases} \quad (4)$$

The other approach is Matching based Fitness (MF) which relies on how much a learned skill can cover the subspace of a raw skill in target task. Agent constructs its CG through living in target task. Then, the *raw skills* are defined by community detection of CG. To find the fitness of each learned skill, agent finds matches of the learned skills to the raw skills by calculating the ratio of their common region to the raw skill's region, as follows:

$$Fitness(O) = \max_{O' \in RawSkills_{Target}} \frac{|O \bigcap O'|}{|O'|} \quad (5)$$

Phase 3. Transferring Relevant Skills and Learning Target Task. The selected skills by previous phase are firstly considered as gestating skills which are allowed to have a gestating period (e.g., 10 episodes), where they cannot be selected for execution but their policies are updated using off-policy learning. Each gestating skill finishing its gestating period would be added to the agent's action set as a learned skill.

4. RESULTS

The proposed method was assessed using a set of well-known 4 dimensional continuous test domain for RL, Pinball domain [1] (see Figure 1). Three transfer learning scenarios are considered to transfer learned skills from an environment as source task to a harder one as target task; 1) from env_1 to env_2, 2) from env_2 to env_3, and 3) from env_3 to env_4.

Table 1 and Figure 3 illustrates the performance of five agents through these scenarios. The agents are: 1) an agent with flat policy (SARSA), 2) a GSL agent, 3) a SCL agent [1], 4) an TF-STL agent employing trajectory based fitness, and 5) an MF-STL agent using matching based fitness. Note that the first three agents do not utilize the transfer learning mechanism.

Four metrics introduced in [8] is used to measure the benefits of transfer (see Table 2). The results indicates that TF-STL outperforms MF-STL in terms of *Jumpstart* metric, because TF-STL transfers only the successful options on the solution path in the source task, whereas MF-STL can transfer any skill whose fitness meets the predefined threshold ($\theta = 0.9$). To make it clear, Figure 2 illustrates the skills in the source and target task.

Table 1: Comparison of agents in terms of average number of steps to reach goal stat

	GSL	MF-STL	TF-STL	SCL	SARSA
env_1	176.7	129.5	147.2	317.6	3105.2
env_2	(±23.0)	(±15.3)	(±22.4)	(±130.7)	(±731.5)
env_2	257.2	198.2	215.8	781.5	10117
env_3	(±33.5)	(±36.3)	(±31.4)	(±265.3)	(±1731.8)
env_3	571.6	455.8	490.1	891.5	23890.9
env_4	(±84.9)	(±47.9)	(±87.7)	(±139.7)	(±5933.8)

Table 2: Comparison of TF-STL and MF-STL methods

		Jumpstart	Asymptotic Performance	Transfer ratio	Time to threshold
env_1	TF	1500(±481)	8262(±352)	1.2%(±0.03)	314(±44)
env_2	MF	2386(±572)	8538(±243)	5.5%(±0.09)	221(±94)
env_2	TF	760(±217)	5336(±112)	0.8%(±0.034)	226(±24)
env_3	MF	1200(±192)	5786(±101)	2.3%(±0.04)	228(±48)
env_3	TF	7539(±3080)	7005(±308)	1.01%(±0.03)	203(±21)
env_4	MF	12047(±6172)	7554(±183)	3.6%(±0.06)	165(±37)

5. CONCLUSION

In this paper, we propose a novel graph based skill acquisition method, named GSL, and a skill based transfer learning framework, named STL. The experimental results indicate the effectiveness of the proposed methods in dealing with continuous reinforcement learning problems. For a broader review of the proposed methods, the reader may refer to its original paper in [5].

REFERENCES

[1] G. Konidaris and A. A. S. Barreto. Skill discovery in continuous reinforcement learning domains using skill chaining. In *Advances in Neural Information Processing Systems*, pages 1015–1023, 2009.

[2] A. Lazaric. Transfer in Reinforcement Learning : a Framework and a Survey. *Reinforcement Learning*, 12:143–173, 2012.

[3] P. Moradi, M. E. Shiri, and N. Entezari. Automatic Skill Acquisition in Reinforcement Learning Agents Using Connection Bridge Centrality. *Communications in Computer and Information Science*, pages 51–62, 2010.

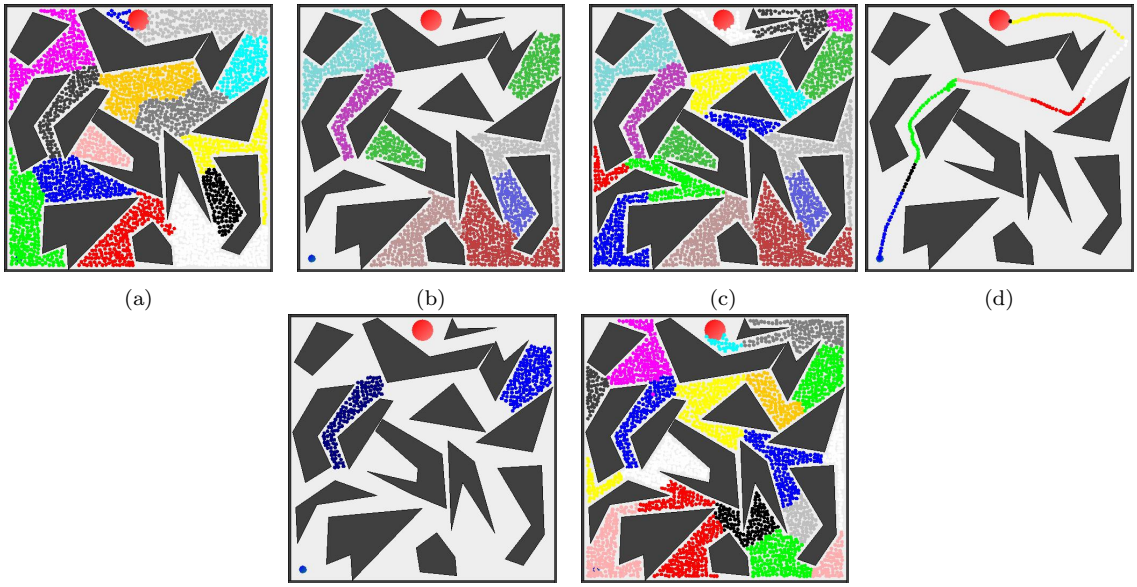
[4] P. Moradi, M. E. Shiri, A. A. Rad, A. Khadivi, and M. Hasler. Automatic skill acquisition in reinforcement learning using graph centrality measures. *Intelligent Data Analysis*, 16:113–135, 2012.

[5] F. Shoeleh and M. Asadpour. Graph based skill acquisition and transfer learning for continuous reinforcement learning domains. *Pattern Recognition Letters*, 2016.

[6] O. Simsek and A. G. Barto. Skill characterization based on betweenness. *Advances in Neural Information Processing Systems 21*, pages 1497–1504, 2008.

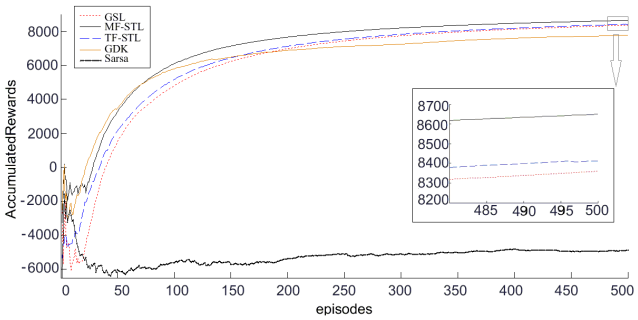
[7] N. Taghizadeh and H. Beigy. A novel graphical approach to automatic abstraction in reinforcement learning. *Robotics and Autonomous Systems*, 61(8):821–835, aug 2013.

[8] M. E. Taylor and P. Stone. Transfer Learning for Reinforcement Learning Domains : A Survey. *Journal of Machine Learning Research*, pages 1633–1685, 2009.

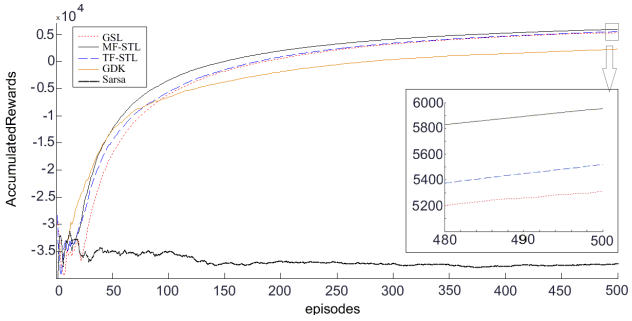


(a) (b) (c) (d) (e) (f)

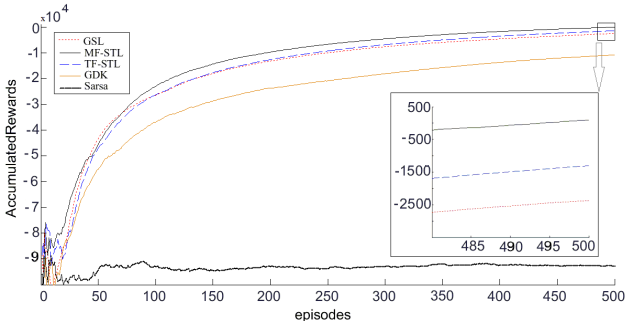
Figure 2: Extracted Skills in source task env_3 (a), Selected Skills in MF-STL method (b), Final Learned Skills in target task env_4 using MF-STL approach (c), a trajectory sample in source task env_3 (d), Selected Skills in TF-STL method (b), Final Learned Skills in target task env_4 using TF-STL approach(f).



(a) Transferring from env_1 to env_2



(b) Transferring from env_2 to env_3



(c) Transferring from env_3 to env_4

Figure 3: Comparison of learning performance of five agents: 1) agent with flat policy (Sarsa), 2) GSL agent with the best configuration, 3) SCL agent, 4) agent employing skill based transfer learning with trajectory based fitness (TF-STL), and 5) agent using skill based transfer learning with matching based fitness (MF-STL), in env_2(a), env_3(b)and env_4(c) environments