Effects of Task Similarity on Policy Transfer with Selective Exploration in Reinforcement Learning

Extended Abstract

Akshay Narayan
School of Computing
National University of Singapore, Singapore
anarayan@comp.nus.edu.sg

Tze Yun Leong
School of Computing
National University of Singapore, Singapore
leongty@comp.nus.edu.sg

ABSTRACT

The SEAPoT algorithm [9] is a knowledge transfer mechanism in model-based reinforcement learning. By constructing subspaces around the changed regions, and selectively and efficiently exploring the target task, the transfer is most effective when the source and target tasks share similar objectives but differ in the transition dynamics. In this work, we identify the similarity between tasks using a new light-weight metric, based on the Jensen-Shannon distance, and show how the degree of similarity affects the transfer efficacy. We also empirically show that SEAPoT performs better in terms of jump starts and average rewards, as compared to the state-of-the-art policy reuse methods.

KEYWORDS

Reinforcement learning; Policy transfer; Transfer in RL; Similarity metric

ACM Reference Format:
Akshay Narayan and Tze Yun Leong. 2019. Effects of Task Similarity on Policy Transfer with Selective Exploration in Reinforcement Learning. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 3 pages.

1 INTRODUCTION

The selective exploration and policy transfer algorithm (SEAPoT) aims to solve a class of problems that can reuse prior knowledge maximally, while adapting to changes when prior knowledge is insufficient. Examples of real world applications include assistive robots in geriatric care homes and hospital wards. We focus on the settings where the source and target tasks differ in the transition dynamics and/or reward functions. We define similarity using the distance between the corresponding state-action transition distributions of the two tasks. The similarity in the environments is captured in a shared state-action space, and the difference is represented in a distribution of environmental elements leading to different transition dynamics.

In SEAPoT, the agent follows the source policy until a change is detected in the environment. Limited exploration is performed in the target task to circumvent the surrounding region of the changed point until a *known state* with respect to the source task is reached. The agent then continues to follow the source task policy from this known state.The approach introduces new ways to identify the changes across tasks and construct subspaces to focus on the most relevant parts of the task's state space to limit the exploration.

2 SEAPOT MECHANISM

The SEAPoT algorithm (introduced in [9]) involves two stages, change detection and selective exploration. We model the values of the state features as a time series and perform change detection using the product partition approach [1]. In selective exploration, the agent constructs a subspace local to the change identified based on its existing knowledge. The subspace construction happens in a breadth-first manner. We call the set of states reachable from any given state, s_i, by taking n actions as the n-step closure, $C^n(s_i)$. The subspace, $M' = \langle S', A', T', R' \rangle$, is a well formed MDP, where $A' \subseteq A$ is the action set, R' is the reward function, and T' is the transition function in this sub-space that must be learned to solve the task. The set of states in M' are identified from the n-step closure of the origin state, $S' = C^n(s_i)$.

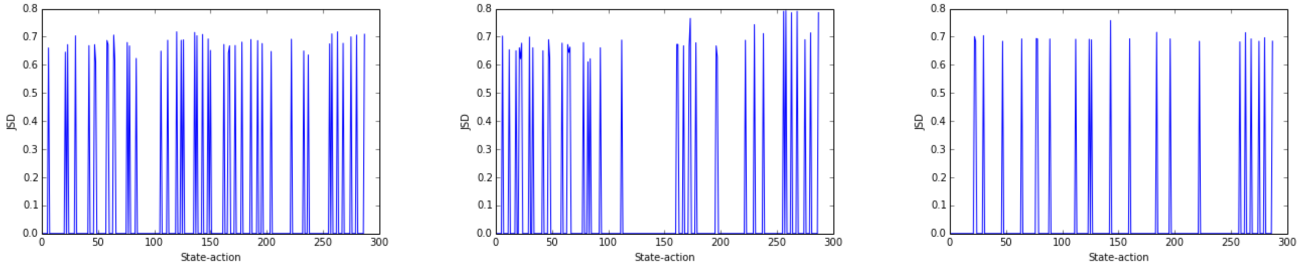
3 TASK SIMILARITY

To prevent or minimize negative transfer, we determine if the source and target tasks are "similar". It is difficult to determine task similarity a priori, unless the transition functions, value functions and/or reward functions for both tasks are available. Prior efforts in determining task similarity require pre-defined task models. Common metrics such as the Kantorovich distance metric [6, 11] and Bisimulation metric [7], are computationally expensive.

We define a new, light weight, metric based on the Jensen-Shannon distance [4] (JSD) to compute task similarity in the problems that share the same state-actions. Jensen-Shannon distance is defined as the square root of the Jensen-Shannon divergence, D_{JS}. JSD is computed as shown below.

$JSD = \sqrt{D_{JS}} = \sqrt{\frac{1}{2}D_{KL}(p,m) + \frac{1}{2}D_{KL}(q,m)}$, where, $m = \frac{p+q}{2}$ and, D_{KL} is the Kullback-Leibler divergence, and p and q are any two probability distributions. The computation complexity of JSD is linear in the number of elements in the two distributions [8]. The task difference ($\Delta_{S,T}$) is calculated as follows: In the shared state-action space, calculate the bin-bin distance, JSD, among the corresponding state-action transition distributions of the two tasks. In the JSD equation, p and q are the transition distributions of the corresponding state-action pairs in the source and target tasks. The distance between each corresponding state-action pair is passed through a step function ($\mathbb{I}$) to determine its contribution to the task difference. The task difference is then the summation of $\mathbb{I}$ over all

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.



(a) $\Delta_{T1,T2}$ (b) $\Delta_{T1,T3}$ (c) $\Delta_{T1,T4}$

Figure 1: Difference between source and target tasks (more spikes mean tasks are much different from each other)

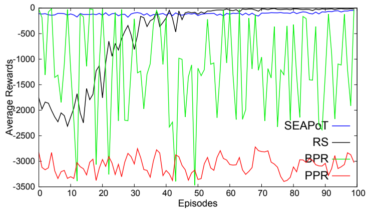


Figure 2: Comparison between SEAPoT, PPR, RS and BPR

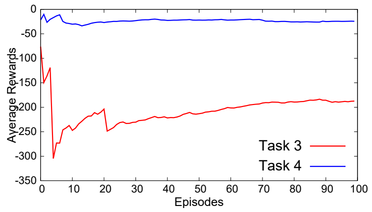


Figure 3: Performance of SEAPoT for two target tasks, $T3$ and $T4$

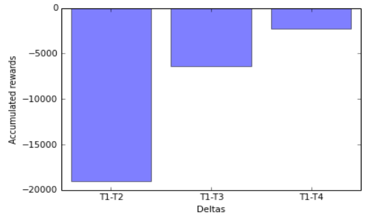


Figure 4: Performance of SEAPoT at different Δ's

the state-action pairs $\Delta_{S,T} = \sum_{(s,a)} \mathbb{I}(JSD)$. In Fig 1, each spike indicates the difference in the probability distribution for a particular state-action pair. The task difference is then calculated according to $\Delta_{S,T}$ above.

We run a few episodes of the target task to obtain an initial estimate of the transition probabilities and use this to determine the task similarity. Intuitively, the task difference metric $\Delta_{S,T}$ identifies the number of locations where the environment has changed from the source task to the target, where the source knowledge is no longer applicable in the target.

4 EXPERIMENTS

We report the experiment results on a taxi [3] like environment (state-space size=8000) and compare our work with three state-of-the-art policy reuse methods [2, 5, 10] (Fig.2). SEAPoT improves on the average rewards over both probabilistic policy reuse and policy transfer using reward shaping. We attribute the performance improvement to the following reasons: (i) The agent reuses its behavioral knowledge learned in the source to the maximum extent in the target task; (ii) there is minimal exploration in the target environment.

Effects of Task Difference on Performance

We hypothesize that the performance of SEAPoT degrades when the source and the target tasks are drastically different. In this experiment we use a simple navigation task in our test-bed to verify the hypothesis. Without loss of generality, assume the tasks $T1$, $T2$, $T3$ and $T4$. $T1$ is the source task and the others are the targets. We plot the differences between each pair of tasks in Figure 1. The spikes indicate the state-action combinations where the two tasks differ; fewer number of spikes mean the tasks are more similar or "closer" to each other. In our experiment, $T4$ is closer to $T1$ than $T2$ (Fig 1) and we expect transferred knowledge to have a more positive impact on learning $T4$. To show the performance difference of SEAPoT at different degrees of similarity, we plot the average rewards of two tasks, $T3$ and $T4$ in Figure 3. The performance degradation is more pronounced in the case of $T2$; it is not shown in the plot for clarity of presentation. Next, we plot the accumulated rewards for each of the target tasks at the end of the learning in Figure 4. As expected, $T2$ suffers from negative transfer whereas $T4$ does better with transfer.

5 DISCUSSION AND FUTURE WORK

In model-based reinforcement learning, the environment models learned in a (source) task can be used to encode valuable information that can be transferred to another (target) task. SEAPoT explores the subspace generated around the changed regions of the target task for policy transfer. Source task policy is reused by exploiting online, local information in the target tasks to adapt to the new setting. We define a new metric to measure the task similarity and examine the effects of similarity on the performance of knowledge transfer.

In future, we plan to exploit the information contained in the learned models to generate better subspaces and reducing the sample complexity of the target task learning. We also plan to use simulation for look-ahead to identify tasks structures and similarities, generate partial policies for reuse in the target task, and design mechanisms to decide when knowledge transfer is not beneficial.

ACKNOWLEDGMENTS

This research was partially supported by Academic Research Grants: MOE2016-T2-2-068 and T1-251RES1827 from the Ministry of Education in Singapore.

REFERENCES

[1] Daniel Barry and John A Hartigan. 1993. A Bayesian analysis for change point problems. *J. Amer. Statist. Assoc.* 88, 421 (1993), 309–319.

[2] Tim Brys, Anna Harutyunyan, Matthew E Taylor, and Ann Nowé. 2015. Policy transfer using reward shaping. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems (AAMAS-15)*. 181–188.

[3] Thomas G Dietterich. 2000. Hierarchical reinforcement learning with the MAXQ value function decomposition. *J. Artif. Intell. Res.(JAIR)* 13 (2000), 227–303.

[4] Dominik Maria Endres and Johannes E Schindelin. 2003. A new metric for probability distributions. *IEEE Transactions on Information Theory* 49, 7 (2003), 1858–1860.

[5] Fernando Fernández, Javier García, and Manuela Veloso. 2010. Probabilistic policy reuse for inter-task transfer learning. *Robotics and Autonomous Systems* 58, 7 (2010), 866–871.

[6] Norm Ferns, Prakash Panangaden, and Doina Precup. 2004. Metrics for finite Markov decision processes. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence (UAI-04)*. AUAI Press, 162–169.

[7] Norm Ferns, Doina Precup, and Sophia Knight. 2014. Bisimulation for markov decision processes through families of functional expressions. In *Horizons of the Mind. A Tribute to Prakash Panangaden*, Franck van Breugel, Elham Kashefi, Catuscia Palamidessi, and Jan Rutten (Eds.). Springer, 319–342.

[8] Yuma Iwasaki, A Gilad Kusne, and Ichiro Takeuchi. 2017. Comparison of dissimilarity measures for cluster analysis of X-ray diffraction data from combinatorial libraries. *npj Computational Materials* 3, 1 (2017), 4.

[9] Akshay Narayan, Zhuoru Li, and Tze-Yun Leong. 2017. SEAPoT-RL: Selective Exploration Algorithm for Policy Transfer in RL. In *Thirty first AAAI Conference on Artificial Intelligence (AAAI-17)*. 4975–4977.

[10] Benjamin Rosman, Majd Hawasly, and Subramanian Ramamoorthy. 2016. Bayesian policy reuse. *Machine Learning* (2016), 1–29. doi: 10.1007/s10994-016-5547-y.

[11] Jinhua Song, Yang Gao, Hao Wang, and Bo An. 2016. Measuring the distance between finite markov decision processes. In *Proceedings of the 2016 International Conference on Autonomous Agents and Multiagent Systems (AAMAS-16)*. International Foundation for Autonomous Agents and Multiagent Systems, 468–476.