Case-based Policy Inference

Ruben Glatt, Felipe Leno da Silva, and Anna Helena Reali Costa
Escola Politécnica da Universidade de São Paulo, Brazil
{ruben.glatt, f.leno, anna.reali}@usp.br

ABSTRACT

This paper introduces the Case-based Inference approach to Reinforcement Learning so as to reuse knowledge from previously solved tasks. We propose the *Case-based Policy Inference (CBPI)* algorithm that accelerates learning by selecting similar source tasks from a library of tasks with their respective policies to solve a new target task. In our experiments we show that by blending the selected policies during training, we achieve significantly faster convergence to the optimal policy.

Keywords

Reinforcement Learning, Policy Reuse, Case-based Inference

1. INTRODUCTION

Although Reinforcement Learning (RL) [12] has been successfully used to autonomously learn how to solve complex tasks [14, 10], it takes a relative long time to get to good solutions. Agents applying RL techniques are known to require a large number of samples of interactions with the environment to infer an effective policy even for simple tasks. In order to alleviate this sample complexity, the Transfer Learning [13, 11] community has been devoting much effort to reuse knowledge from previously learned tasks and progressively reduce the sample complexity of learning new tasks.

We here propose to follow a *Case-based Reasoning* (CBR) approach [1], which describes the methodology to reuse existing knowledge in a general manner [16]. In the RL setting, one can adapt the *Case-based Inference* (CBI) framework [7, 2] to take advantage of policies from previously learned RL tasks and accelerate the learning process in a new target task. This makes sense because by using a stochastic approach for the behaviour selection of the learning agent, one can get rid of restrictions in regard to the used algorithm for training the source task policies and also different reward functions that could exist for different tasks. We here describe a method, termed *Case-based Policy Inference (CBPI)*, to build a library of task solutions, which we use to select relevant source policies in order to improve the training of a new target task. Our experimental evaluation shows a clear speed-up of *CBPI* while learning a new task reusing existing knowledge compared to learning from scratch.

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.*

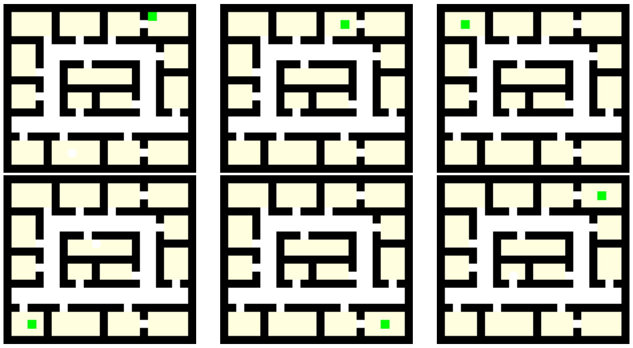

Figure 1: The used gridworld domain and the individual tasks (different goal positions). First row: Ω, Ω_1, Ω_2. Second row: Ω_3, Ω_4, Ω_5

.

Because of the space restrictions we only discuss two of the most related works here. The first one is Probabilistic Policy Reuse [5], where the authors propose a framework to autonomously build a library of policies for a given domain. During training this library can be used to accelerate the learning of a new task. The approach differs from ours in a way that we are blending policies all the time during training according to usefulness for the current task, where the other approach selects whole policies for a certain time to train with that one policy. The second approach uses generalization of knowledge to build a single abstract policy [8] and shows that abstracting over states and actions can be beneficial to knowledge transfer. Instead of combining the existing policies according to their suitability for the task this approach builds a general policy once from existing knowledge whereas our approach builds a generalization on-the-fly for each specific task according to performance in the target task.

2. CASE-BASED POLICY INFERENCE STRATEGY

To establish the connection to the CBR methodology we formulate our algorithm (Algorithm 1) following the general CBR cycle [9]:

1. **RETRIEVE:** For every new task Ω an agent analyses the existing knowledge and builds a temporary library

$\mathcal{L}_{Train}$ of policies that are expected to be most useful from the source library $\mathcal{L}$ and also adds the new policy Π_Ω. For example, in our *Gridworld* domain (see Figure 1), we determine the task similarity as the euclidean distance between goal states and limit the selection by a percentage of the grid diagonal. Although primitive, this method already provides good results in this domain.

2. **REUSE:** During training of the new task we then use the selected policies and blend their action-values by transforming them into action probabilities $pa_{\Omega_i}(a_j|s)$ using a *softmax* function [3] and multiplying them with their respective policy probability $pp_{\Omega_i}(\Omega)$. We then add up each weighted action probability of each policy to get the final action probability distribution for the current state $pa_\Omega(a_j|s)$.
3. **REVISE:** The policy probability is re-evaluated in regular intervals by performing a number of evaluation runs and using the resulting average steps in another *softmax* function to get the probability distribution over the selected policies. While we are using the described policy blending mechanism to make our action selection, we are updating only the target policy during training.
4. **RETAIN:** At the end of the training process the new policy gets added to the policy library.

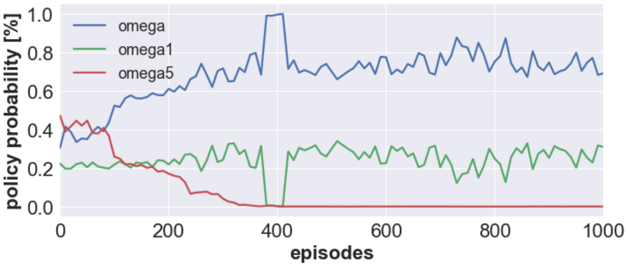

Figure 2: Development of policy probabilities $pp_{\Omega_i}(\Omega)$ for policies in the training library $\mathcal{L}_{Train}$, Π_{Ω_1} and Π_Ω, and for the target policy Π_Ω while learning the target task Ω.

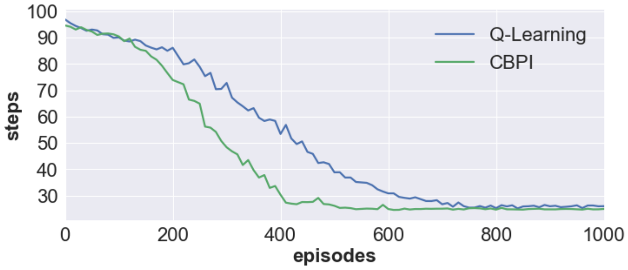

Figure 3: Direct comparison between vanilla Q-Learning and our CBPI algorithm.

3. EXPERIMENT AND RESULTS

The experiment we describe serves to show the viability of our approach when reusing knowledge in a simple domain. An agent has to find its way to a goal state in a *Gridworld* domain consisting of rooms and connecting corridors (see Figure 1). For our purposes we let the agent train on five source tasks (Ω_{1-5}) and save the learned policies in our policy library $\mathcal{L}$. Then the agent has to train on a new target task Ω and learn a policy Π_Ω for that task.

We train every task for 1000 episodes, where an episode ends, when the goal state is reached or a maximum of 100 steps is performed. The results here are averages over 10 runs per task. As a baseline we train the target task from scratch using Q-Learning [15] and compare it with our *CBPI* algorithm, where we provide the policy library and the algorithm selects the most suitable policies autonomously.

Algorithm 1 Case-based Policy Inference (CBPI)

```
Require: Library $\mathcal{L}$ of source tasks $\Omega_s$ and policies $\Pi_s$
Require: Target task $\Omega$
Initialize target policy $\Pi_\Omega$
RETRIEVE:
Select most similar tasks for training library $\mathcal{L}_{Train}$
Add target task $\Omega$ and policy $\Pi_\Omega$ to $\mathcal{L}_{Train}$
Set policy probabilities $pp_{\Omega_i}(\Omega)$ for policies in $\mathcal{L}_{Train}$
repeat  (for $e$ training episodes)
    REUSE:
    Get action probabilities $pa_{\Omega_i}(a_j|s)$ at each step
    Blend policies: $pa_\Omega(a_j|s) = \sum_i pa_{\Omega_i}(a_j|s) * pp_{\Omega_i}(\Omega)$
    Select probabilistic action
    REVISE:
    Update $\Pi_\Omega$
    Update $pp_{\Omega_i}(\Omega)$ only on defined interval
until training ends
RETAIN:
Add new policy $\Pi_\Omega$ to policy library $\mathcal{L}$
```

In Figure 2 we can see the development of the policy probabilities over training episodes. At first the policies have a very similar probability but soon the current policy becomes dominant and has the greatest influence on the training performance. In Figure 3 we can see the benefits of our *CBPI* algorithm over Q-Learning, the average steps per evaluation interval converge much faster to an optimal value that is lower as with Q-Learning even at the end of training.

4. CONCLUSIONS

In this short paper, we introduced the idea of using CBI for knowledge transfer in RL by reusing learned policies. We proposed the CBPI algorithm and showed the potential of the approach in a simple experiment. In the future, we want to build on this concept and extend its applicability to more complex domains and empirically evaluate it against other transfer methods as for example Probabilistic Policy Reuse [5]. We also intend to investigate the feasibility of this approach to use for knowledge transfer in Deep Reinforcement Learning architectures [4, 6] and how to integrate it in an intelligent network structure.

Acknowledgments

We are grateful for the support from CAPES, CNPq (grant 311608/2014-0), and FAPESP (grants 2015/16310-4, 2016/21047-3). We also thank Google (Research Award) and the Nvidia corporation (GPU donation).

REFERENCES

[1] A. Aamodt and E. Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications*, 7(1):39–59, 1994.

[2] M. Anthony and J. Ratsaby. A probabilistic approach to case-based inference. *Theoretical Computer Science*, 589:61–75, 2015.

[3] J. S. Bridle. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In *Proceedings of the 2nd International Conference on Neural Information Processing Systems (NIPS)*, pages 211–217. MIT Press, 1989.

[4] Y. Du, G. V. de la Cruz, Jr., J. Irwin, and M. E. Taylor. Initial progress in transfer for deep reinforcement learning algorithms. In *Proceedings of Deep Reinforcement Learning: Frontiers and Challenges Workshop*, New York City, NY, USA, July 2016.

[5] F. Fernández and M. Veloso. Probabilistic policy reuse in a reinforcement learning agent. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS)*, pages 720–727, 2006.

[6] R. Glatt, F. L. d. Silva, and A. H. R. Costa. Towards knowledge transfer in deep reinforcement learning. In *Brazilian Conference on Intelligent Systems (BRACIS)*, pages 91–96, 2016.

[7] E. Hullermeier. Credible case-based inference using similarity profiles. *IEEE Transactions on Knowledge and Data Engineering*, 19(6):847–858, 2007.

[8] M. L. Koga, V. Freire, and A. H. Costa. Stochastic abstract policies: Generalizing knowledge to improve reinforcement learning. *Cybernetics, IEEE Transactions on*, 45(1):77–88, 2015.

[9] J. Kolodner. *Case-based reasoning*. Morgan Kaufmann, 2014.

[10] V. Mnih, D. Silver, A. A. Rusu, M. Riedmiller, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[11] S. J. Pan and Q. Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, 2010.

[12] R. S. Sutton and A. G. Barto. *Introduction to reinforcement learning*. MIT Press, Cambridge, MA, USA, 1998.

[13] M. E. Taylor and P. Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research (JMLR)*, 10:1633–1685, 2009.

[14] G. Tesauro. Temporal difference learning and td-gammon. *Communications of the ACM*, 38(3):58–68, 1995.

[15] C. J. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8(3-4):279–292, 1992.

[16] I. Watson. Case-based reasoning is a methodology not a technology. *Knowledge-based systems*, 12(5):303–308, 1999.