Action Categorization for Computationally Improved Task Learning and Planning

Extended Abstract

Lakshmi Nair
Georgia Institute of Technology
lnair3@gatech.edu

Sonia Chernova
Georgia Institute of Technology
chernova@cc.gatech.edu

ABSTRACT

This paper explores the problem of task learning and planning, contributing the *Action-Category Representation (ACR)* to improve computational performance of both Planning and Reinforcement Learning (RL). ACR is an algorithm-agnostic, abstract data representation that maps objects to action categories (groups of actions), inspired by the psychological concept of *action codes*. We validate our approach in StarCraft and Lightworld domains; our results demonstrate several benefits of ACR relating to improved computational performance of planning and RL, by reducing the action space for the agent.

KEYWORDS

Cognitive Psychology; Reinforcement Learning; Planning

ACM Reference Format:
Lakshmi Nair and Sonia Chernova. 2018. Action Categorization for Computationally Improved Task Learning and Planning. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 3 pages.

1 INTRODUCTION

Recent fMRI studies show that the human brain uses *action codes* – automatically evoked memories of prototypical actions that are related to a given object – to bias or constrain expectation on upcoming manipulations [4]. For instance, a knife and an apple seen together evoke the action codes of “cutting apple with knife” and “peeling apple with knife”.

In this work we use action codes extracted from human demonstrations of the task or self-exploration by the agent in order to construct the *Action Category Representation (ACR)*: an algorithm-agnostic, abstract data representation that encodes a mapping from objects to action categories (groups of actions) for a task. Specifically, we use action codes to build action categories that help improve computational performance in both task planning and reinforcement learning by constraining the action space for the agent.

Most of the existing literature targeted towards computational improvements in planning and RL have focused on representations that are often specific to the approach (either planning or RL) and the formalisms therein [5, 6, 10]. Other approaches that have previously used human demonstrations as a means of constraining the action space tend to be sensitive to the number and/or optimality of

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

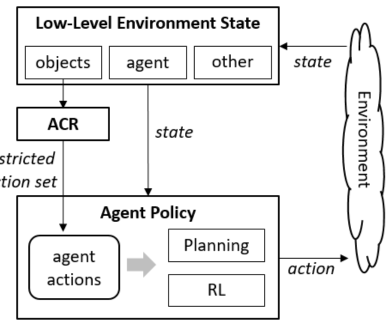

Figure 1: Objects in the low-level environment state are mapped via ACR to action categories to restrict the action set used in the planning or RL techniques

the demonstrations provided [1, 7, 9]. ACR addresses these limitations by encompassing abstract knowledge regarding object-action relationships in the form of what the agent *can* do, rather than what the agent *should*. Hence, the novelty of ACR is three-fold:

(1) contains and appropriately represents algorithm-agnostic information (action categories) for planning, as well as RL, to improve their computational performance,
(2) minimizes agent-object interactions for learning the action associations of a new object (not seen during demonstration or exploration); and
(3) requires one or few human demonstrations and is robust to the optimality of these demonstrations.

2 ACR

ACR is constructed by grouping actions into categories using action codes. An action code can be formally represented as:

$$((o_1, o_2 ... o_j), (a_1, a_2 ... a_k))$$

Where $(o_1, ... o_j)$ represents a set of objects and $(a_1, ... a_k)$ represents the set of actions associated with them for the task. For instance, the action code corresponding to the knife and apple example above is $((apple, knife), (peel, cut))$. We formulate the problem of constructing ACR as a bipartite graph partitioning problem from the set of objects O to the set of actions A. We group sets of actions that are associated with the same set of objects into a single group called an "Action Category", denoted by A^c, thus resulting in a reduced many-to-one bipartite graph that is the source of the computational improvements of ACR (figure 2). The construction of ACR is an online and incremental process since new objects and action categories can be easily incorporated by adding/removing corresponding edges in the bipartite graph.

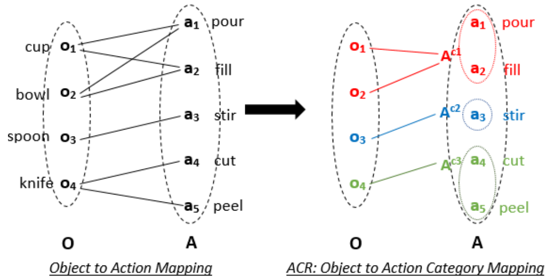

Figure 2: Bipartite graphs representing the mapping from objects to actions (left), and objects to action categories (right)

In addition to computational improvements, ACR helps reduce number of agent-object interactions (A_{obj}) required to learn the action associations of new objects (not seen during self-exploration or demonstration). This is achieved using information entropy as an action selection mechanism. Entropy allows choosing the action that provides maximum information about the category membership of an object, thus reducing the total number of agent-object interactions required to learn all its action associations.

3 RESULTS

We conducted experiments in two domains: Starcraft for planning and Lightworld [2] for RL to study the computational improvements of ACR when combined with each approach.

In Starcraft, we compared the number of agent-object interactions (A_{obj}) for ACR and a baseline approach (without action categorization) to learn the actions associated with a set of previously unseen game objects. Fewer A_{obj} is computationally preferred since it makes the learning or planning faster. Results shown in table 1, demonstrate the benefits of ACR in reducing A_{obj} compared to the baseline approach. This is also beneficial in applications such as real-world robotic manipulation tasks, where there is an implicit time or cost constraint associated with each agent-object interaction. We also compared performance of classical planning to ACR-based planning to show computational performance improvements due to the reduced action space (figure 3).

In the Lightworld domain, we studied the effect of number and optimality of user demonstrations on ACR as well as the computational benefits of integrating ACR with RL, specifically Q-learning. The ACR was constructed from a single user demonstration and was compared to Human-Agent Transfer (HAT) [8], an approach that summarizes user demonstrations into a decision list for improving the learning performance of RL algorithms. As shown in figure 4, ACR is robust to number and optimality of user demonstrations compared to HAT. While HAT outperforms ACR given sufficient expert demonstrations, ACR is a more beneficial approach given non-expert users and fewer demonstrations.

4 CONCLUSIONS

To conclude, we presented the Action-Category Representation that allows online categorization of objects to action categories based on action codes. Our results demonstrate some of the key benefits of ACR in terms of reduced action space resulting from the action groupings, computational improvements when used with planning and RL, and reduced demonstration requirements with robustness to demonstration errors.

While the domains described here are discrete in nature, ACR is also applicable to continuous domains by discretizing the state space into states where interaction with an object is possible/not possible. For instance an object may be interacted with, if the agent is within a certain distance of it. Approaches such as [3] have discussed discretization of continuous state spaces for RL and in this manner, ACR can also be extended to continuous domains.

Civilization	Number of objects explored	Total A_{obj} w/o categorization	Total A_{obj} w ACR
Terrans	4	36	10
Protoss	7	63	21
Zergs	9	81	28

Table 1: ACR reduces number of agent-object interaction (A_{obj}) compared to baseline

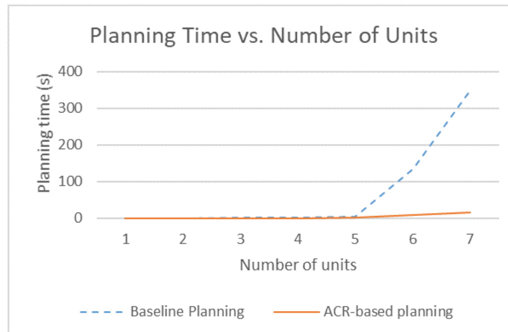

Figure 3: Graph showing effect of number of Dragoons on planning time

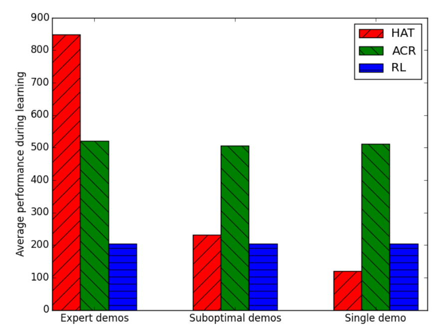

Figure 4: Summary of the average learning performances based on number and quality of demonstrations for HAT and ACR. Q-learning (RL) also shown for comparison.

5 ACKNOWLEDGEMENT

This work is supported by NSF IIS 1564080.

REFERENCES

[1] Tim Brys, Anna Harutyunyan, Halit Bener Suay, Sonia Chernova, Matthew E Taylor, and Ann Nowé. 2015. Reinforcement Learning from Demonstration through Shaping.. In *IJCAI*. 3352–3358.

[2] George Konidaris and Andrew G Barto. 2007. Building Portable Options: Skill Transfer in Reinforcement Learning.. In *IJCAI*, Vol. 7. 895–900.

[3] Jonathan Mugan and Benjamin Kuipers. 2008. Continuous-domain reinforcement learning using a learned qualitative state representation. (2008).

[4] Ricarda I Schubotz, Moritz F Wurm, Marco K Wittmann, and D Yves von Cramon. 2014. Objects tell us what action we can expect: dissociating brain areas for retrieval and exploitation of action knowledge during action observation in fMRI. *Frontiers in psychology* 5 (2014).

[5] Mohan Sridharan, Michael Gelfond, Shiqi Zhang, and Jeremy Wyatt. 2015. A refinement-based architecture for knowledge representation and reasoning in robotics. *arXiv preprint arXiv:1508.03891* (2015).

[6] Mohan Sridharan and Ben Meadows. 2017. An Architecture for Discovering Affordances, Causal Laws, and Executability Conditions. *Advances in Cognitive Systems* 5 (2017), 1–16.

[7] Halit Bener Suay, Tim Brys, Matthew E Taylor, and Sonia Chernova. 2016. Learning from demonstration for shaping through inverse reinforcement learning. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 429–437.

[8] Matthew Edmund Taylor, Halit Bener Suay, and Sonia Chernova. 2011. Using Human Demonstrations to Improve Reinforcement Learning.. In *AAAI Spring Symposium: Help Me Help You: Bridging the Gaps in Human-Agent Collaboration*.

[9] Andrea Lockerd Thomaz, Cynthia Breazeal, et al. 2006. Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance. In *Aaai*, Vol. 6. 1000–1005.

[10] Kadir Firat Uyanik, Yigit Caliskan, Asil Kaan Bozcuoglu, Onur Yürüten, Sinan Kalkan, and Erol Sahin. 2013. Learning Social Affordances and Using Them for Planning.. In *CogSci*.