Object Exchangeability in Reinforcement Learning

Extended Abstract

John Mern
Stanford University
Stanford, California
jmern91@stanford.edu

Dorsa Sadigh
Stanford University
Stanford, California
dorsa@stanford.edu

Mykel Kochenderfer
Stanford University
Stanford, California
mykel@stanford.edu

ABSTRACT

Although deep reinforcement learning has advanced significantly over the past several years, sample efficiency remains a major challenge. Careful choice of input representations can help improve efficiency depending on the structure present in the problem. In this work, we present an *attention-based* method to project inputs into an efficient representation space that is invariant under changes to input ordering. We show that our proposed representation results in a search space that is a factor of $m!$ smaller for inputs of m objects. Our experiments demonstrate improvements in sample efficiency for policy gradient methods on a variety of tasks. We show that our representation allows us to solve problems that are otherwise intractable when using naïve approaches.

KEYWORDS

Knowledge Representation; Reasoning; Reinforcement Learning

1 INTRODUCTION

Deep reinforcement learning (RL) has achieved state-of-the-art performance across a variety of tasks [7, 10]. However, successful deep RL training requires large amounts of sample data. Various learning methods have been proposed to improve sample efficiency, such as model-based learning and incorporation of Bayesian priors [4, 11].

The key insight of this paper is that we can significantly improve efficiency by leveraging the exchangeable structure inherent in many reinforcement learning problems. That is, for a state space that can be factored into sets of sub-states, presenting the factored state in a way that does not rely on a particular ordering of the sub-states can lead to significant reduction in the search-space.

In this work, we propose an attention mechanism as a means to leverage object exchangeability. We propose a mechanism that is permutation invariant in that it will produce the same output for any permutation of the items in the input set and show that this representation reduces the input search space by a factor of up to $m!$, where m is the number of exchangeable objects.

2 BACKGROUND AND RELATED WORK

Deep RL is a class of methods to solve Markov Decision Processes (MDPs) using deep neural networks. Solving an MPD requires finding a policy π that maps all states in a state-space $\mathcal{S}$ to an action to maximize total accumulated rewards.

Formally, we can define an *object* in an MDP to be a subset of the state space that defines the state of a single entity in the problem environment. In an aircraft collision avoidance problem, an object could be defined by the values associated with a single aircraft. It is well known that as the number of objects grow, the size of the MDP search space grows exponentially [8].

For MDPs with exchangeable objects, an optimal policy should provide the same action for any permutation of the input. When states are represented as ordered sets, as is common, this must be learned by the policy during training.

Many methods have been proposed instead to enforce this by permutation invariant input representations. The Object Oriented MDP (OO-MDP) framework uses object-class exchangeability to represent states in an order-invariant space for discrete spaces [3]. Approximately Optimal State Abstractions [1] proposes a theoretical approximation to extend OO-MDP to continuous domains. Object-Focused Q-learning [2] uses object classes to decompose the Q-function output space, though it does not address the input.

Deep Sets [14] proposes a permutation invariant abstraction method to produce input vectors from exchangeable sets. The method proposed produces a static mapping. That is each input object is weighted equally regardless of value during the mapping.

Our method improves upon Deep Sets by introducing an attention mechanism to dynamically map the inputs to the permutation-invariant space. Attention mechanisms are used in various deep learning tasks to dynamically filter the input to a down-stream neural network to emphasize most important parts of the original input [5, 6, 13]. We adapt a mechanism from recent work in natural language processing, which use a dot-product neural layer to efficiently apply dynamic attention [12].

3 PROBLEM FORMALIZATION

Our objective is to propose an attention mechanism that will take sets of objects S_i as inputs and produce abstractions S_i^* such that the mapping is permutation invariant. This output can then be used as an input to the policy neural network in an RL problem (e.g. a deep Q-Network or action policy). Our hypothesis is that learning using this abstract representation will be more sample efficient than learning on the original object set.

We propose the attention network architecture shown in fig. 1, which is a permutation invariant implementation of dot-product attention. For a single input set $S_i \leftarrow \{s_i^{(1)}, \ldots, s_i^{(m)}\}$, the m object
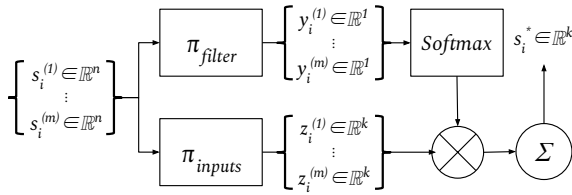
Figure 1: Permutation invariant attention mechanism. - Objects $\{s_i^{(1)}, \ldots, s_i^{(m)}\}$ are passed into neural networks $\pi_{\mathbf{inputs}}$ and $\pi_{\mathbf{filters}}$ with a Softmax. The outputs are multiplied and summed along the object dimension for the final order-invariant output s_i^*.
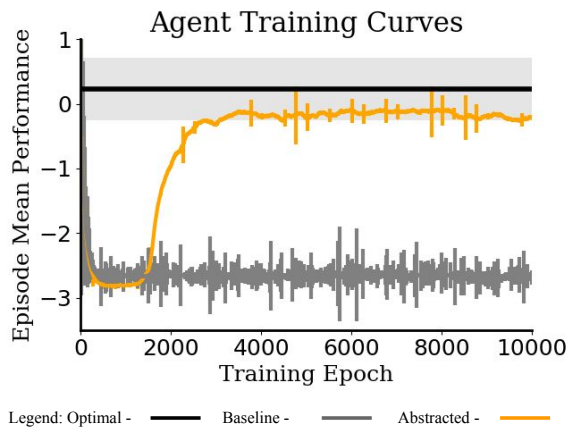

Figure 2: Convoy Task Training Curve

state vectors are individually passed through feed-forward neural networks π_{filter} and π_{inputs}. The scalar outputs of the filter graph are concatenated into a single vector $y_i \in \mathbb{R}^m$ and the *softmax* operation is applied. These outputs are then multiplied element-wise by the concatenated outputs of the network π_{inputs}. In this way, the output of π_{filter} acts as the attention filter, weighting the inputs by importance prior to summation. The elements of the weighted vector z_i are then summed over the m different objects, resulting in a single vector $s_i^* \in \mathbb{R}^k$. This s_i^* vector is then used as the input to the policy neural network.

We can now define bounds on the sample efficiency benefits of an invariant mapping. Define a state space $\mathcal{S}$ such that $\{s_1, \ldots, s_m\} \in \mathcal{S}$, where m is the number of objects. Let each object s_i take on n unique values. Representing the states as *ordered* sets of s_i results in a state-space size $|\mathcal{S}|$ that can be calculated from the expression for m permutations of n values. If all objects are exchangeable, there exists an abstraction that is permutation invariant. Since the order does not matter, the size of this abstract state $|\hat{\mathcal{S}}|$ can then be calculated from the expression for m combinations of n values.

$$|\mathcal{S}| = \frac{n!}{(n-m)!}, \quad |\hat{\mathcal{S}}| = \frac{n!}{m!(n-m)!} \tag{1}$$

Using permutation invariant representation reduces the input space by a factor of $\frac{|\mathcal{S}|}{|\hat{\mathcal{S}}|} = \frac{1}{m!}$ compared to ordered set representation.

It can be shown that it is necessary and sufficient for a mapping f to be invariant on all countable sets $\mathcal{X}$ if and only if it can be decomposed using transformations ϕ and ρ, where ϕ and ρ are any vector valued functions to the form [14]:

$$f(X) = \rho\Big(\sum_{x \in \mathcal{X}} \phi(x) \Big) \tag{2}$$

It can be shown that the proposed attention mechanism may be decomposed to the above form to prove it is permutation invariant. For problems with multiple classes of exchangeable objects, a separate attention mechanism can be deployed for each class.

4 EXPERIMENTS AND RESULTS

We conducted a series of experiments to validate the effectiveness of our proposed abstraction. The first two tasks are simple MDPs in which a scavenger agent navigates a continuous two-dimensional world to find *food* particles.

In the scavenger tasks, the state space contains vectors $s \in \mathbb{R}^{2m+2}$, where m is the number of target objects. The vector contains the relative position of each food particle as well as the ego position of the agent. The agent receives a reward of $+1.0$ when reaching a food particle, and a reward of -0.05 for every time-step otherwise. The episode terminates upon reaching a food particle or when the number of time-steps exceeds a limit. Scavenger Task 2 introduced *poison* particles in addition to the food particles (one poison for each food particle). If an agent reaches a poison particle, a reward of -1.0 is given and the episode terminates.

The third task is a convoy protection task with variable numbers of objects. The task requires a defender agent to protect a convoy that follows a predetermined path through a 2D environment. Attackers are spawned at the periphery of the environment during the episode, and the defender must block their attempts to approach the convoy. The state space is the space of vectors representing the state of each non-ego object in the environment. The episode terminates when all convoy members either reach the goal position or are reached by an attacker.

For each task, we trained a set of policies with the attention mechanism as well as a baseline policies that use a standard ordered set to represent the input space. Each policy was trained with Proximal Policy Optimization (PPO) [9], policy-gradient algorithm.

For each scavenger task, we trained a policy for on tasks having one to five food particles. The baseline policies were unable to achieve optimal performance for tasks with more than two food particles in either scavenger task. The policy trained with our attention mechanism was able to learn an optimal policy for all cases with no increase in the number of required training samples. For the convoy task, the abstracted policy approached optimal behavior after approximately 2,500 epochs, where the baseline policy showed no improvement after 10,000 epochs, as shown in fig. 2.

These experiments demonstrate the effectiveness of the proposed approach to enhance the scalability of the PPO policy gradient learning algorithm. Together, these experiments validate our hypothesis that leveraging object exchangability for input representation can improve the efficiency of deep reinforcement learning.

REFERENCES

[1] David Abel, D. Ellis Hershkowitz, and Michael L. Littman. 2016. Near Optimal Behavior via Approximate State Abstraction. In *International Conference on Machine Learning (ICML)*.

[2] Luis C. Cobo, Charles L. Isbell, and Andrea L. Thomaz. 2013. Object Focused Q-learning for Autonomous Agents. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

[3] Carlos Diuk, Andre Cohen, and Michael L. Littman. 2008. An Object-oriented Representation for Efficient Reinforcement Learning. In *International Conference on Machine Learning (ICML)*.

[4] Shixiang Gu, Timothy Lillicrap, Ilya Sutskever, and Sergey Levine. 2016. Continuous Deep Q-learning with Model-based Acceleration. In *International Conference on Machine Learning (ICML)*.

[5] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. 2015. Spatial transformer networks. In *Advances in Neural Information Processing Systems (NIPS)*.

[6] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* (2015).

[7] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. 2013. Playing Atari with Deep Reinforcement Learning. *Nature* 518 (2013), 529–533.

[8] Philipp Robbel, Frans Oliehoek, and Mykel Kochenderfer. 2016. Exploiting Anonymity in Approximate Linear Programming: Scaling to Large Multiagent MDPs. In *AAAI Conference on Artificial Intelligence (AAAI)*.

[9] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. *CoRR* abs/1707.06347 (2017). arXiv:1707.06347 http://arxiv.org/abs/1707.06347

[10] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. 2017. Mastering the game of Go without human knowledge. *Nature* 550 (2017), 354–359.

[11] Benjamin Spector and Serge J. Belongie. 2018. Sample-Efficient Reinforcement Learning through Transfer and Architectural Priors. *CoRR* abs/1801.02268 (2018). arXiv:1801.02268 http://arxiv.org/abs/1801.02268

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems (NIPS)*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). 5998–6008.

[13] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning (ICML)*.

[14] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan R Salakhutdinov, and Alexander J Smola. 2017. Deep Sets. In *Advances in Neural Information Processing Systems (NIPS)*.