Multi-Source Policy Aggregation in Heterogeneous and Private Environmental Dynamics

Mohammadamin Barekatain*
Technical University of Munich
Munich, Germany
m.barekatain@tum.de

Ryo Yonetani
OMRON SINIC X
Tokyo, Japan
ryo.yonetani@sinicx.com

Masashi Hamaya
OMRON SINIC X
Tokyo, Japan
masashi.hamaya@sinicx.com

1. Introduction

We envision a future scenario where robotic agents working in diverse and private environments help a new agent in an unknown environment to learn its policy efficiently. For instance, imagine various types of pick-and-place robotic agents working in a factory. While the agents are involved in the same task, dynamics of the environment in which the task is performed is different based on each robot's kinematics (*e.g.*, degree of freedom, link length, and joint orientations) and dynamics (*e.g.*, joint damping, armature, and friction) [1]. Moreover, no prior knowledge about the dynamics of environments as well as the specification of agents policies can be available for a new agent due to the confidentiality of products and processes in the factory. Other relevant scenarios include autonomous vehicles on private land and home assistants interacting with people privately.

The problem setting shown above makes it hard to adopt many existing approaches for efficient learning of a target agent's policy. For instance, meta-learning approaches typically require an agent to be trained on a diverse task distribution [4], which is not possible here due to the privacy of environments. Also, existing transfer learning approaches that focus on the transfer of policies between dynamics, require prior information about the environments [1] or policy configuration (*e.g.*, actor network weights and agent's value function [2]) which are both unavailable.

In such scenarios, we argue that the target agent can get information from other private agents through their policies (hereafter *source* policies) that act as a black-box function mapping states to actions. Specifically, we propose a new sample efficient approach named *MULTI-source POLicy AggRegation (MULTIPOLAR)*. Much like a multipolar neuron that can integrate information coming from other neurons, our MULTIPOLAR aggregates the actions produced by the source policies to serve a robust baseline action. It also learns an additional policy to predict a 'residual' around the baseline actions to mitigate the unseen dynamics of the target agent's environment. As a result, MULTIPOLAR achieves training sample efficiency since the aggregation of source actions provides a strong inductive bias.

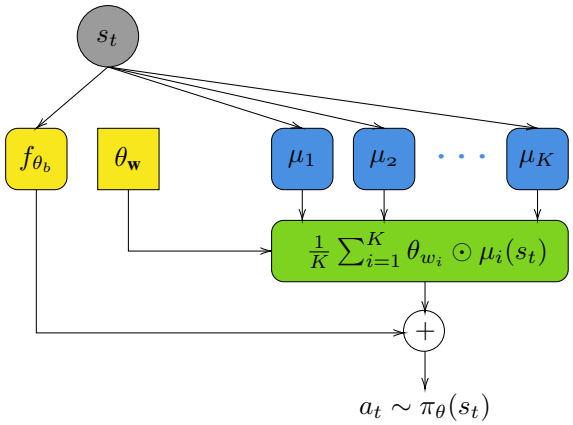

Figure 1: The proposed MULTIPOLAR policy network.

As a preliminary experiment, we evaluate MULTIPOLAR with two public simulated environments with continuous and discrete action spaces: Roboschool Hopper[1] and OpenAI Acrobot[2]. Our experimental results demonstrate that MULTIPOLAR allows a new agent to learn its policy significantly faster on average compared to when it is trained from scratch.

2. Proposed Method

Preliminaries We formulate our policy aggregation problem under the standard Reinforcement Learning (RL) framework, where an agent interacts with its environment modeled by a Markov Decision Process (MDP). An MDP is represented by a 6-tuple $(\rho_0, \gamma, \mathbb{S}, \mathbb{A}, R, T)$ where ρ_0 is the initial state distribution and $\gamma \in (0, 1]$ is the discount

*Work done as an intern at OMRON SINIC X

[1]https://github.com/openai/roboschool
[2]https://gym.openai.com/envs/Acrobot-v1

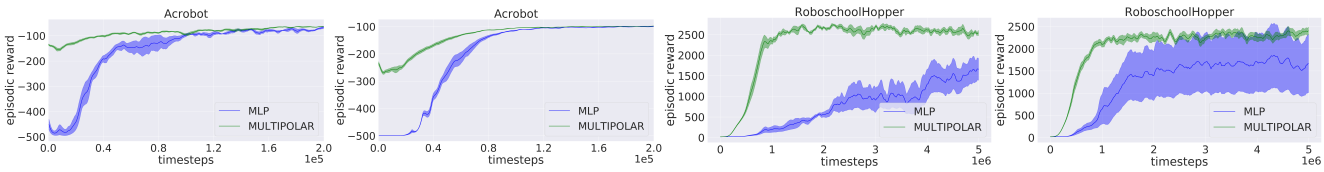

Figure 2: Example learning curves for Acrobot and Hopper averaged over trials. The shaded area represents 1 standard error.

factor. At each timestep t, given the current state $s_t \in \mathbb{S}$, the agent executes an action $a_t \in \mathbb{A}$ based on its policy $\pi_\theta(a_t|s_t)$ which is parameterized by θ. The environment returns a reward $r_t = R(s_t, a_t)$ and transitions to the next state $s_{t'}$ based on the environment state transition distribution $T(s_{t'}|s_t, a_t)$. In this framework, RL aims to maximize the expected return with respect to the policy parameters θ. In this work, we use ε to denote a particular environment such as Hopper which the agent is interacting with and ε_i to denote different unknown environmental dynamics for that particular environment ε. Specifically, $\rho_0, \gamma, \mathbb{S}, \mathbb{A}, R$ are the same for ε_is but the state transition distribution $T_i(s_{t'}|s_t, a_t)$ is different.

Problem statement Given an environment ε and a library (set) of K deterministic source polices $L = \{\mu_1, \mu_2, ..., \mu_K\}$, where each μ_i is acquired from ε_i, quickly learn an optimal policy π_θ for ε_{target} by exploiting knowledge from L. Each source policy μ_i can be parameterized (*e.g.* learned from interacting with an environment) or non-parameterized (*e.g.*, heuristically designed by humans). Either way, we assume no prior knowledge about the representations of μ_i (*e.g.* their network architectures) as well as their environmental dynamics and we only have access to their predicted deterministic actions $a_{i,t} = \mu_i(s_t)$. Moreover, μ_i's are not necessarily optimal on the ε_i they were acquired from.

Policy aggregation Our proposed MULTIPOLAR policy network π_θ, illustrated in Figure 1, aims at improving sample efficiency of training an agent on a particular environment with unknown dynamics ϵ_{target} by leveraging knowledge from L. The main idea is to learn how useful each action of each source policy is for ϵ_{target} by learning a state-independent scale vector θ_{w_i} for each source policy μ_i. To ensure the optimality of MULTIPOLAR policy, a residual policy f_{θ_b} is learned around the average of scaled source actions. Specifically, the mean actions at timestep t is:

$$u_\theta(s_t) = f_{\theta_b}(s_t) + \frac{1}{K}\sum_{i=1}^{K} \theta_{w_i} \odot \mu_i(s_t) \qquad (1)$$

where $\theta = \{\theta_b, \theta_{w_1}, \ldots, \theta_{w_K}\}$. $u_\theta(s_t)$ is the estimated mean of a multivariate Gaussian distribution (for continuous action space environments) or a categorical distribution (for discrete action space environments) and final actions are sampled from the corresponding distribution. MULTIPOLAR can be trained with any model-free policy gradient methods given that we can take the gradient of policy performance with respect to θ.

3. Experiments

We aim to empirically demonstrate the importance of aggregating source policies for achieving training sample efficiency. We evaluate the effectiveness of MULTIPOLAR with four source policies ($K = 4$) on Hopper with continuous action space and Acrobot with discrete action space. In both environments we compare MULTIPOLAR policy to the standard MultiLayer Perceptron (MLP) policy network typically used in RL. To have a fair comparison, both baseline MLP and MULTIPOLAR are trained with Proximal Policy Optimization algorithm [3] with the same hyperparameters and same network size for MLP and f_{θ_b}.

Results In both Hopper and Acrobot experiments, we designed 100 target environments by randomly sampling the environmental dynamics parameters such as links length and mass. For each target environment, we trained MULTIPOLAR three times with different sets of source policies acquired from random ϵ_i's. Each training was done three times with different random seeds. Figure 2 shows the learning curves of MULTIPOLAR policy (averaged over 3 choices of source policies $\times$ 3 random seeds $=$ 9 trials) and that of the baseline policy (averaged over the same 3 random seeds) for different ϵ_{target}'s. It clearly shows that MULTIPOLAR outperforms the baseline in terms of sample efficiency and sometimes the final episodic reward. It is also noteworthy that MULTIPOLAR learning curves are significantly more consistent across different trials, given that their standard errors are much smaller than the baseline.

References

[1] T. Chen, A. Murali, and A. Gupta. Hardware Conditioned Policies for Multi-Robot Transfer Learning. In *NeurIPS*, 2018. 1

[2] F. L. Da Silva and A. H. R. Costa. A Survey on Transfer Learning for Multiagent Reinforcement Learning Systems. *J. Artif. Int. Res.*, 64(1):645–703, 2019. 1

[3] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 2

[4] J. Vanschoren. Meta-Learning: A Survey. *arXiv preprint arXiv:1810.03548*, 2018. 1