Guided Exploration in Reinforcement Learning via Monte Carlo Critic Optimization

Extended Abstract

Igor Kuznetsov
Independent Researcher
Oxford, United Kingdom
igorkuznetsov14@gmail.com

ABSTRACT

In reinforcement learning (RL), the class of deep deterministic off-policy algorithms is effectively applied to solve challenging continuous control problems. Current approaches commonly utilize random noise as an exploration method, which has several drawbacks, including the need for manual adjustment for a given task and the absence of exploratory calibration during the training process. We address these challenges by proposing a novel guided exploration method that uses an ensemble of Monte Carlo Critics for calculating exploratory action correction. The proposed method enhances the traditional exploration scheme by dynamically adjusting exploration. Subsequently, we present a novel algorithm that leverages the proposed exploratory module for both policy and critic modification. The presented algorithm demonstrates superior performance compared to modern RL algorithms across a variety of problems in the DMControl suite.

KEYWORDS

Reinforcement Learning; Exploration; Off-policy; Monte Carlo

ACM Reference Format:
Igor Kuznetsov. 2024. Guided Exploration in Reinforcement Learning via Monte Carlo Critic Optimization: Extended Abstract. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 3 pages.

1 INTRODUCTION

In reinforcement learning (RL), exploration methods can be broadly categorized as undirected or directed [8]. Undirected methods involve generating random exploratory actions based on a desired exploration-exploitation trade-off, while directed algorithms rely on information provided by a policy or a learned world model.

In the context of continuous control problems, the class of deep deterministic off-policy methods has gained high popularity in the RL community due to its implementation simplicity and state-of-the-art results. From an exploration perspective, algorithms such as DDPG [6] and TD3 [3] utilize Gaussian or time-dependent Ornstein-Uhlenbeck noise applied to deterministic actions. While serving as a simple and effective exploration tool, methods based on random noise face several limitations, such as the need for manual adjustment for a given task and the absence of exploratory calibration during the training process.

We address the mentioned weaknesses of random exploration by introducing a differentiable module capable of guiding a policy in a purposeful exploratory direction. Specifically, we advocate using an ensemble of Q-function approximations trained to predict Monte Carlo Q-values as this exploratory module. Through optimizing multiple independent predictions and utilizing the action gradient derived from calculated variance, we can obtain an uncertainty estimate for a given state. Using this gradient, we introduce an exploratory action correction aimed at exploring the least traversed regions of the environment. We demonstrate that our proposed exploration outperforms other exploration methods across a range of continuous control tasks.

Additionally, we introduce a novel algorithm that leverages the proposed uncertainty-based module for both the actor and critic components of the model architecture. Our proposed method demonstrates superior results when compared to modern RL algorithms on a set of tasks from the DMControl suite [7]. The source code is available at github.com/schatty/MOCCO.

This work is licensed under a Creative Commons Attribution International 4.0 License.

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

2 PRELIMINARIES

We consider a standard RL setup, in which an agent interacts with an environment $\mathcal{E}$ at discrete time steps aiming to maximize the reward signal. The environment is a Markov Decision Process (MDP) that can be defined as $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \rho, \gamma \rangle$, where $\mathcal{S}$ is a state space, $\mathcal{A}$ is an action space, $\mathcal{R}$ is a reward function, ρ is a transition dynamics and $\gamma \in [0, 1]$ is a discount factor. At time step t the agent receives state $s_t \in \mathcal{S}$ and performs action $a_t \in \mathcal{A}$ according to policy π, a distribution of a given s that leads the agent to the next state s_{t+1} according to the transition probability $\rho(s_{t+1}|s_t, a_t)$. After providing the action to $\mathcal{E}$, the agent receives a reward $r_t \sim \mathcal{R}(s_t, a_t)$. The discounted sum of rewards during the episode is defined as a *return* $R_t = \sum_{i=t}^{T} \gamma^{i-t} r(s_i, a_i)$.

The RL agent aims to find the optimal policy π_θ, with parameters θ, which maximizes the expected return from the initial distribution $J(\theta) = \mathbb{E}_{s_i \sim \rho_\pi, a_i \sim \pi_\theta}[R_0]$. The action-value function Q is at core of many RL algorithms and denotes the expected return when performing action a from the state s following the current policy π: $Q^\pi(s, a) = \mathbb{E}_{s_i \sim \rho_\pi, a_i \sim \pi}[R_t|s, a]$.

3 METHOD OF GUIDED EXPLORATION

The proposed agent consists of two components: an actor-critic part to provide a deterministic policy and an exploratory module that

Table 1: A comparison of exploration approaches. Average episodic score from 10 trials. Each trial is a mean of the last 10 episodes.

Task	*NO EXPL*	*NORM*	*OU*	*RND*	*GE*
acrobot	0.65	0.59	1.67	0.67	**150.46**
point_mass	726.80	709.90	605.09	743.36	**785.25**
pendulum	247.411	359.94	215.13	241.13	**792.33**
walker-walk	921.80	940.80	953.13	881.04	**955.977**
walker-run	644.32	596.71	622.06	589.50	**677.73**
hopper-stand	56.40	34.86	58.67	51.91	**602.14**
hopper-hop	18.30	27.18	16.31	14.14	**167.12**
human-walk	3.32	79.16	30.02	129.83	**329.25**

adjusts the policy output action to facilitate directed exploration. The deterministic policy $\pi_\theta(s)$ is parameterized by θ and optimizes reward maximization to produce a *base* action a^{b}. The exploratory module EM optimizes auxiliary intrinsic objective to produce an *exploratory* action correction a^{e}. The policy and the exploratory module are jointly optimized to produce an additive action for collecting transitions for off-policy updates:

$$a := a^{\mathrm{b}} + a^{\mathrm{e}}, a^{\mathrm{b}} \sim \pi_\theta(s) \tag{1}$$

For the actor-critic part, we use the TD3 algorithm [3] as a backbone.

The exploratory module $\mathrm{EM}_\omega(s|\theta)$ is parameterized by ω and conditioned on policy parameters θ. The proposed controller features an ensemble of Q-function approximators $\{q_1(s,a), ..., q_n(s,a)\}$ that predict the collected Monte Carlo Q-values. The values for the update are sampled from a small experience replay buffer $\mathcal{D}_{MC}$ with recent trajectories collected by the policy. Given n ensemble predictions, the controller EM estimates prediction uncertainty as the ensemble disagreement:

$$\mathrm{EM}_\omega(s|\theta) := \mathrm{Var}\left(\{q_1(s,a), ..., q_n(s,a))\}\right), \tag{2}$$

where $q_i(s,a) = \mathbb{E}_{s\sim\mathcal{D}_{MC}, a\sim\pi_\theta}[R], i \in [1..n]$. This disagreement reflects both epistemic uncertainty of controller parameters and aleatory uncertainty of the collected Monte Carlo Q-values. During optimization, controller's objective is to reduce uncertainty in a supervised fashion by minimizing square distance between the predicted and collected returns:

$$J_\phi = \sum_{i=1}^{n} (Q^{MC}(s,a) - q_i(s,a))^2, \tag{3}$$

where (s,a) is a state-action pair sampled from $\mathcal{D}_{MC}$ with the corresponding Monte Carlo Q-value $Q^{MC}(s,a)$. Taking the gradient of the controller w.r.t. action, we obtain the direction towards maximizing the uncertainty under current controller parameters ω:

$$\nabla_a \mathrm{EM}_\omega = \nabla_a \mathrm{Var}\left(\{q_1(s,a), ..., q_n(s,a))\}\right)|_{a\sim\pi_\theta(s)}. \tag{4}$$

The gradient value is then normalized with gradient norm and scaled with running mean and standard deviation for each action component separately producing exploratory action a^{e}.

Table 1 shows the results for 8 control tasks from DMControl Suite. The studied exploration methods are as following: actions are sampled without noise (NO EXPL), with Normal Gaussian noise (NORM), temporally correlated noise drawn from an Ornstein-Uhlenbeck process (OU), curiosity-based exploration with RND method from [1], and with proposed guided exploration method (GE). Results demonstrate higher rewards of guided exploration over the conventional approaches for all environments.

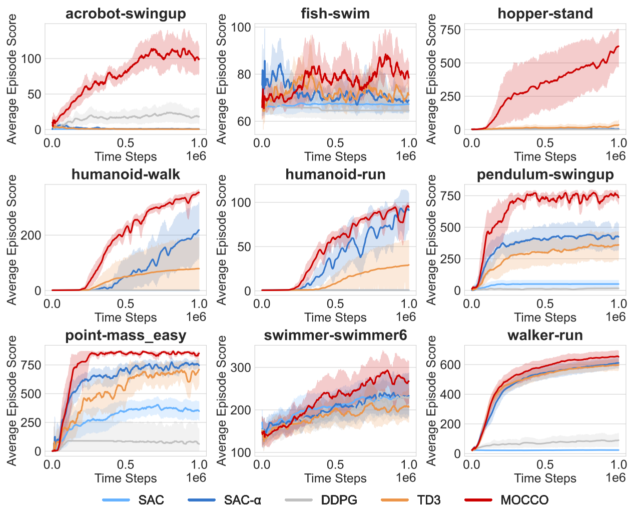

Figure 1: Learning curves of evaluation algorithms. The results are averaged across 10 random seeds.

4 MOCCO ALGORITHM

Based on the presented technique of guided exploration, we propose a novel deep RL algorithm **Mo**nte **C**arlo **C**ritic **O**ptimization (MOCCO) that incorporates an ensemble of Monte Carlo critics not only for the actor, as an exploratory module, but also for the critic to alleviate Q-value overestimation. Several works have studied Q-value overestimation in continuous control setting [2, 4, 5]. Here, we propose to use a mean from the ensemble of Monte Carlo values as a second pessimistic Q-value estimate during critic optimization, resulting in the following critic's objective:

$$J_Q = (Q - Q')^2 + \beta \cdot (Q - Q^{MC})^2, \tag{5}$$

where $Q^{MC} = \mu(\{q_1, ..., q_n\})$, and β is a coefficient controlling an impact of the pessimistic estimate. For the given state-action pair (s,a) the critic estimate $Q(s,a)$ is generally greater than the corresponding Monte Carlo estimate Q^{MC}, as the latter predicts estimates of the past sub-optimal policies. In RL, Monte Carlo estimates have high variance and low bias, whereas one-step TD methods have less variance but can be biased. Here, we combine both methods during Q-function optimization.

Practically, the MOCCO algorithm is based on TD3 with the following differences: (1) it uses guided exploration during action selection; (2) it does not use second TD critic; (3) the mean of MC-critics ensemble is used during the critic optimization. The comparative results between the MOCCO algorithm and modern RL algorithms is depicted in Figure 1. MOCCO outperforms other approaches, sometimes with significant margins, e.g. for acrobot, pendulum, and hopper domains.

REFERENCES

[1] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. 2018. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894* (2018).

[2] Kamil Ciosek, Quan Vuong, Robert Loftin, and Katja Hofmann. 2019. Better exploration with optimistic actor-critic. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. 1787–1798.

[3] Scott Fujimoto, Herke Hoof, and David Meger. 2018. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*. PMLR, 1587–1596.

[4] Arsenii Kuznetsov, Pavel Shvechikov, Alexander Grishin, and Dmitry Vetrov. 2020. Controlling overestimation bias with truncated mixture of continuous distributional quantile critics. In *International Conference on Machine Learning*. PMLR, 5556–5566.

[5] Igor Kuznetsov and Andrey Filchenkov. 2021. Solving Continuous Control with Episodic Memory. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, Zhi-Hua Zhou (Ed.). International Joint Conferences on Artificial Intelligence Organization, 2651–2657. Main Track.

[6] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015).

[7] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. 2018. Deepmind control suite. *arXiv preprint arXiv:1801.00690* (2018).

[8] Sebastian B Thrun. 1992. Efficient exploration in reinforcement learning. (1992).