Discrete off-policy policy gradient using continuous relaxations

Andre Cianflone
Mila - McGill University

Zafarali Ahmed
Mila - McGill University

Riashat Islam
Mila - McGill University

Avishek Joey Bose
Mila - McGill University

William L. Hamilton
Mila - McGill University

Abstract

Off-Policy policy gradient algorithms are often preferred to on-policy algorithms due to their sample efficiency. Although sound off-policy algorithms derived from the policy gradient theorem exist for both discrete and continuous actions, their success in discrete action environments have been limited due to issues arising from off-policy corrections such as importance sampling. This work takes a step in consolidating discrete and continuous off-policy methods by adapting a low-bias, low-variance continuous control method by relaxing a discrete policy into a continuous one. This relaxation allows the action-value function to be differentiable with respect to the discrete policy parameters, and avoids the importance sampling correction typical of off-policy algorithms. Furthermore, the algorithm automatically controls the amount of relaxation, which results in implicit control over exploration. We show that the relaxed algorithm performs comparably to other off-policy algorithms with less hyperparameter tuning.

Keywords: policy gradient, off-policy actor-critic, continuous relaxation

Acknowledgements

Z.A. is funded by a Canada Graduate Scholarship, A.C. is funded by a Borealis AI Fellowship.

1 Introduction

Policy gradient methods are a class of algorithms used to solve reinforcement learning problems (RL) by directly optimizing a parameterized policy. *On-policy learning* uses data collected from this policy to compute gradient updates. Despite being rather successful [1], they can be sample inefficient as new data needs to be collected for each gradient update. Consequently, *Off-policy learning* is preferred due to its ability to re-use data collected from older policies. In particular, off-policy methods support data re-use from multiple behaviour policies, while learning a desired target policy.

While algorithms such as the Deep Deterministic Policy Gradient (Deep DPG) [2] exist for environments with continuous actions, there has not been much progress for discrete actions due to the lack of a viable discrete reparameterization approach. Algorithms like off-policy actor critic (Off-PAC) [3] and Actor Critic with Experience Replay (ACER) [4] can be derived for discrete action environments. However, the reliance on the importance sampling corrections limits their use in practice due to its high variance gradient estimate [5]. Recent work introduces Actor-Critic with Emphatic Weightings (ACE) [6] as another approach to discrete action off-policy learning that introduces the first "off-policy policy gradient theorem". However, ACE also requires estimating corrections and has not yet been demonstrated in more complex domains.

Our work aims to use successful continuous control algorithms [7] for discrete action environments by using *continuous relaxations* of samples from a discrete policy [8]. In essence, we convert the learning of a discrete policy into a continuous control problem. A particularly interesting side effect of the relaxation is the introduction of a temperature parameter, τ, that controls the amount of relaxation: The temperature can be automatically tuned [9], thereby controlling the entropy of the policy and eliminating the need for external exploration noise. We call this approach a Autotuned, Relaxed, Reparameterized Discrete Domain algorithm (AR2D2). Our contributions are:

1. Using continuous relaxations of discrete categorical samples [10, 8] to find the gradient of the action-value function, resulting in an algorithm similar to DPG [11].
2. Automatic control of the relaxation allowing sufficient exploration and eventual recovery of the optimal policy using a novel objective that balances variance reduction [9] and action-value maximization.

2 Background

We start by covering the off-policy reinforcement learning setting before considering the continuous relaxation in Section 2.1. Consider a Markov decision process $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}, \gamma \rangle$ where $\mathcal{S}$ is a set of states, $\mathcal{A}$ is a set of discrete actions, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the state-transition probabilities, and $\gamma \in [0, 1]$ is the discount factor. The expected discounted return from a state, s_0, is given by the value function: $V^\pi(s) = \mathbb{E}_\pi[\sum_t \gamma^t r_t | s_0 = s]$. In policy gradient methods, we search for a parameterized *target* policy, π_θ, that maximizes $J(\theta) = \sum_s d_{\pi_\theta}(s) V^{\pi_\theta}(s)$ where $d_{\pi_\theta}(s)$ is the stationary state distribution under the policy π_θ.

However, in case of off-policy learning the samples are drawn from the state distribution under the *behaviour policy*, $\mu(s|a)$. Therefore, we optimize $J(\theta) = \sum_s d_\mu(s) V^{\pi_\theta}(s)$ where $d_\mu(s)$ is the stationary state distribution under μ.

Importance sampling techniques (IS) can be used to correct for the discrepancy in the behaviour and target policies [5]. However, IS corrections, being high variance, often make algorithms such as Off-PAC [3] difficult to use in practice. Alternatively, we can use the deterministic policy gradient theorem [12] to avoid IS corrections by considering deterministic policies, $\pi_\theta(a|s) = a$. In particular, DPG proposes a variant of Q-learning for policy gradients, where instead of taking a greedy policy improvement, we can directly improve the policy in the direction of the gradient of the action-value function: $\nabla_\theta J(\theta) = \sum_s d_\mu(s) \nabla_\theta Q^{\pi_\theta}(s, \pi_\theta(s))$. One limitation of the DPG is that it requires differentiable samples.

Differentiable reparameterizations exist for continuous distributions like the Gaussian [13, 14] and have been applied to continuous control problems in RL [11, 7]. While relaxing a categorical distribution has been explored in reinforcement learning as a action-depentent control variate [9] and a policy [15], it has not been fully developed into a viable alternative to well-known algorithms such as DQN [16].

2.1 Continuous Relaxations for Discrete Variables

In this section, we cover background material related to discrete reparmetrization of categorical distributions [10, 8]. Consider the general objective of optimizing parameters θ of a probability distribution, p_θ, to maximize the function f. The gradient is defined as $\nabla_\theta L(\theta) = \nabla_\theta \, \mathbb{E}_{z \sim p_\theta}[f(z)]$. When f is not differentiable the log-deriviative identity[1] can be applied to obtain the REINFORCE estimator, $\nabla_\theta L(\theta) = \mathbb{E}_{z \sim p_\theta(z)}[f(z) \nabla_\theta \log p_\theta(z)]$ which can be estimated using Monte-Carlo sampling [17].

[1]The log-derivative identity is $\nabla x = x \nabla \log x$

In cases where f is differentiable and p_θ can be *reparameterized* through a deterministic function, $z = g(\epsilon, \theta)$, a low variance gradient estimate can be be computed by shifting the stochasticity from the distributional parameters to a standardized noise model, ϵ [13, 14]. Specifically, we can rewrite the gradient computation as $\nabla_\theta L(\theta) = \mathbb{E}_\epsilon[\nabla_g f(g(\epsilon, \theta))\nabla_\theta g(\epsilon, \theta)]$.

The Gumbel-Max trick [18] offers such a reparameterization for the categorical distribution: $z = \arg\max_i[g_i + \log \eta_i]$ where η_i are the log probabilities for a Categorical distribution and g_i are independent and identically distributed noise variables from the Gumbel$(0, 1)$ distribution. While such a reparametrization shifts the distribution parameters to a deterministic node, it introduces a non-differentiable $\arg\max$. The Gumbel-Softmax (GS) [10] distribution proposes to replace the $\arg\max$ with a softmax and temperature parameter τ:

$$y_i = \frac{\exp((\log \eta_i + g_i)/\tau)}{\sum_{j=1}^{k} \exp((\log \eta_j + g_j)/\tau)} \tag{1}$$

where $\tau \to 0$ recovers the $\arg\max$, and $\tau \to \infty$ recovers the uniform distribution. Due to the relaxation, the softmax operation is differentiable providing continuous differentiable samples from this distribution. Computing $\arg\max_i y_i$ corresponds to sampling from a categorical distribution and allows execution in a reinforcement learning environment.

3 Off-Policy Policy Gradients with Gumbel Reparameterization

In this section we discuss how to introduce the Gumbel-Softmax as an alternate parameterized policy for discrete actions in the off-policy setting. We will do this by deriving the gradient for the action-value function to do policy improvement by following the grading direction.

Recall the off-policy learning setup, where the goal is to learn a target parameterized policy π_θ while collecting data from a behaviour policy μ. Consider the gradient of the action-value function:

$$\nabla_\theta J(\pi_\theta) = E_{s\sim d_\mu(s)}[\nabla_a Q(s, a)\nabla_\theta \pi_\theta(s)] \tag{2}$$

Like in DPG, $a = \pi_\theta(s)$, where π_θ is implemented using a Gumbel-Softmax policy with a relaxation parameter, τ (Equation 1). These relaxed discrete actions allow us to take gradients of Q w.r.t. the policy parameters θ, effectively back-propagating through the sampling process. To execute actions in the environment, continuous samples from relaxed policies are discretized using $\arg\max$ so that they correspond to samples from a categorical distribution. In supervised learning tasks, the Gumbel-Softmax temperature parameter is decayed [10] to reduce the relaxation over time. In reinforcement learning, premature annealing may lead to a suboptimal deterministic policy as the policy would fail to sample a diverse number of trajectories (i.e. reduced exploration). The temperature, τ, must be carefully controlled to prevent this outcome. In this work we consider τ is learned during optimization.

We now describe an off-policy actor-critic algorithm we call AR2D2 . We first describe the standard critic update to learn Q, and then describe how the actor parameters are updated. Finally, we discuss how the trainable relaxation parameter is automatically tuned in our setup for exploration and variance minimization. The algorithm is summarized in Algorithm 1.

3.1 Critic Update

We use a Q function parameterized by w, where in our case w are the parameters of a neural network. Unlike DPG, our policy is discrete and allows the Q function to be updated by minimizing the mean squared error (MSE) between Q and a fixed target. To address overestimation bias [19] in the critic update, we employ Double Clipped Q-Learning [7]. The target Q in the MSE now consists of taking the minimum of two Q-functions in the critic update:

$$L(w) = \frac{1}{N}\sum_i (r_i + \gamma \min_{i=1,2} Q'_{\tilde{w}_i}(s_{i+1}, \pi_\theta(s_{i+1})) - Q_w(s_i, a_i))^2 \tag{3}$$

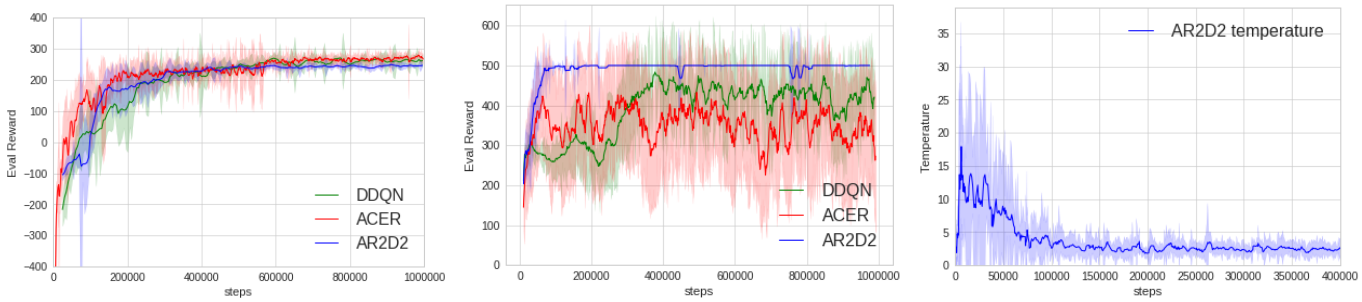
where (s_i, a_i, r_i, s_{i+1}) are a collection of experiences from the environment.

3.2 Actor Update

Expanding the Gumbel-Softmax policy definition from Equation 1 reveals three sets of variables: the categorical probabilities $\{\eta_1, \ldots, \eta_{|A|}\}$, the Gumbel noise $\{g_1, \ldots, g_{|A|}\}$ and the temperature parameter τ. The categorical parameters are implemented with a deep neural network and updated by following the gradient in Equation 2.

3.3 Temperature Update

While the addition of the temperature is added to Gumbel-Softmax as a requirement for being differentiable, its introduction offers a unique opportunity in the reinforcement learning domain to automatically control the balance between



(a) Performance on LunarLander-v2 (b) Performance on CartPole-v1 (c) Temperature auto-tuned in training

Figure 1: **Evaluation performance for three algorithms on (a) LunarLander-v2 and (b) CartPole-v1. (c) shows the behaviour of temperature during learning.** (a) While all three algorithms solve LunarLander (> 200 reward), AR2D2 displays lower variability between random seeds. (b) While ACER and DDQN show high variance for CartPole, AR2D2 converges quickly even when using the same hyperparameters as Lunar Lander. We plot a smoothed mean-return along with standard deviation (shaded) for 5 random seeds. (c) The temperature increases at the start of learning and decays automatically over time.

exploration and exploitation: adjusting the temperature from zero to infinity interpolates the distribution between a deterministic $\arg\max$ and a uniform distribution. The role of τ is then similar to the downstream impact of entropy regularization in that it controls policy stochasticity [20, 21] and avoids the need for external exploration noise often added to off-policy algorithms [16, 2, 19]. Given τ is part of the same computational graph which optimizes η, we formulate two separate gradient updates for τ, one to maxizime discounted return and another to minimize variance.

In order to stabilize the changing policy, we minimize the policy gradient variance w.r.t. τ, as proposed in [9], who optimize the temperature of a relaxed hard threshold control variate. The gradient of the variance in gradient wrto τ is formulated as:

$$\frac{\partial}{\partial \tau}\text{Var}(g(\pi_\eta)) = \frac{\partial}{\partial \tau}\left(\mathbb{E}[g(\pi_\eta)^2] - \mathbb{E}[g(\pi_\eta)]^2\right) = \mathbb{E}\left[2g(\pi_\eta)\frac{\partial g(\pi_\eta)}{\partial \tau}\right] \tag{4}$$

where $g(\pi_\eta)$ is the gradient of Equation 2 w.r.t. the categorical parameters η. The second update takes a gradient ascent step w.r.t. τ in the direction which maximizes Q. Combining the two, we get the following update for τ:

$$\tau_{t+1} = \tau_t + \alpha_\tau^Q[\nabla_a Q(s,a)|_{a=\pi_\theta(s)}\nabla_\tau \pi_\theta(s)] - \alpha_\tau^\sigma \nabla_\tau \text{Var}(g(\pi_\theta)) \tag{5}$$

where α_τ^Q and α_τ^σ are respective learning rates for the gradien updates to maximize Q and minimize the gradient variance from Equation 4. We find that a high α_τ^Q learning rate, allowing the policy to quickly interpolate between exploration and exploitation, is key to quick convergence of the algorithm. The algorithm is summarized in Algorithm 1.

4 Experimental Results

In this section we show the viability of using continuous relaxations by comparing with two state-of-the-art off-policy RL algorithms: Double Deep-Q Learning (DDQN) [19] and ACER [4] on two discrete action environments, LunarLander-v2 and CartPole-v1 [22]. Algorithms are compared based on rollouts of the greedy policy. We tune hyperparameters on LunarLander-v2 and transfer them without modification to CartPole-v1.

All methods solve LunarLander-v2 (Figure 1a). Interestingly, the hyperparameters for AR2D2 found on LunarLander-v2 transferred without modification onto CartPole-v1 unlike ACER and DDQN (Figure 1b) for which we had to fine-tune the learning rate. This suggests that the auto-tuning mechanism might provide increased stability and robustness to hyperparameters in our algorithm. Additionally, we note the remarkable stability of AR2D2 on CartPole (Figure 1b compared to ACER and DDQN, despite the simplicity of the domain. A more robust algorithm would be a strong addition to the current repertoire of RL algorithms and a further exposition of robustness will be left to future work.

In Figure 1c we can see how the temperature parameter increases substantially at the beginning of training, while quickly decreasing around 100,000 steps. An increase in the temperature τ suggests both increased exploration and smoothing of the problem early on during training.

5 Conclusion

In this work, we have shown empirical evidence for using continuous relaxations of discrete random variables in an off-policy policy gradient algorithm. Particularly interesting is the dual purpose of the temperature parameter τ. It controls

both the relaxation and the data collected from the environment, i.e. exploration. Specifically, the relaxation can be seen as a form of smoothing and its relationship to entropy regularization will be explored in future work [21].

In summary, our work has unified discrete and continuous actions in the same off-policy policy gradient algorithm. We expect that other RL algorithms that have previously faced the "differentiabiliy" requirement can successfully take advantage of the relaxation. Future work will consider a more thorough theoretical and empirical investigation of performance as well as the robustness of AR2D2 to hyperparameters.

References

[1] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, 2016.

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

[3] Thomas Degris, Martha White, and Richard S Sutton. Off-policy actor-critic. *arXiv preprint arXiv:1205.4839*, 2012.

[4] Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Remi Munos, Koray Kavukcuoglu, and Nando de Freitas. Sample efficient actor-critic with experience replay. *International Conference on Learning Representations*, 2017.

[5] Doina Precup. Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, page 80, 2000.

[6] Ehsan Imani, Eric Graves, and Martha White. An off-policy policy gradient theorem using emphatic weightings. In *Advances in Neural Information Processing Systems*, 2018.

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

[8] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *International Conference on Learning Representations*, 2017.

[9] George Tucker, Andriy Mnih, Chris J Maddison, John Lawson, and Jascha Sohl-Dickstein. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. In *Advances in Neural Information Processing Systems*, 2017.

[10] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *International Conference on Learning Representations*, 2017.

[11] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International Conference on Machine Learning*, 2014.

[12] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International Conference on Machine Learning*, 2014.

[13] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[14] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *31st International Conference on Machine Learning*, 2014.

[15] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, 2017.

[16] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 2015.

[17] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 1992.

[18] Emil Julius Gumbel. Statistical theory of extreme values and some practical applications. *NBS Applied Mathematics Series*, 1954.

[19] Hado V Hasselt. Double q-learning. In *Advances in Neural Information Processing Systems*, 2010.

[20] Ronald J Williams and Jing Peng. Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, 1991.

[21] Zafarali Ahmed, Nicolas Le Roux, Mohammad Norouzi, and Dale Schuurmans. Understanding the impact of entropy on policy optimization. *arXiv preprint arXiv:1811.11214*, 2018.

[22] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

6 Appendix

The algorithm derived from the updates in Section 3 is shown below:

Algorithm 1 AR2D2

Input: critic networks Q_{w_1}, Q_{w_2}, and Gumbel-Softmax actor network π_η and π_τ
Input: update actor step d, temperature learning rate α_τ, update weight β
Initialize target networks $w_1' \leftarrow w_1, w_2' \leftarrow w_2, \theta' \leftarrow \theta$
Initialize replay memory $\mathcal{D}$
for episode $= 1$ **to** M **do**
  Initialize s_t
  for $t = 1$ **to** T **do**
    Select action $a = \pi_\theta(s_t)$
    Discretize action a using $\arg\max$ to obtain $\hat{a}$.
    Observe $(r, s_{t+1}) = \text{env}(\hat{a})$
    Append $\mathcal{D}$ with tuple (s_t, a, r, s_{t+1})
    Sample mini-batch of N transitions (s, a, r, s') from $\mathcal{D}$
    $y \leftarrow \begin{cases} r & \text{for terminal state } s \\ r + \gamma \min_{i=1,2} Q_{\tilde{w}_i}(s', a) & \text{for non-terminal state} s \end{cases}$
    Update critics $w_i \leftarrow \arg\min_{w_i} N^{-1} \sum (y - Q_{w_i}(s, a))^2$
    if $t \text{mod} d$ **then**
      Update policy gradients:
      $\nabla_\eta J(\eta) = N^{-1} \sum \nabla_a Q(s, a)|_{a=\pi_\theta(s)} \nabla_\eta \pi_\theta(s)]$
      $\nabla_\tau J(\tau) = N^{-1} \sum \nabla_a Q(s, a)|_{a=\pi_\theta(s)} \nabla_\tau \pi_\theta(s)]$
      $\nabla_\tau \text{Var}(g(\pi_\theta)) = \mathbb{E}[2g(\pi_\theta) \nabla_\tau g(\pi_\theta)]$
      Update target networks:
      $w_i' \leftarrow \beta w_i + (1 - \beta) w_i'$
      $\eta' \leftarrow \beta \eta + (1 - \beta) \eta'$
      $\tau' \leftarrow \beta \tau + (1 - \beta) \tau'$
    end if
  end for
end for