Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

A Multi-Agent Off-Policy Actor-Critic Algorithm for Distributed Reinforcement Learning

Wesley Suttle [*] **Zhuoran Yang** [**] **Kaiqing Zhang** [***] **Zhaoran Wang** [****] **Tamer Başar** [†] **Ji Liu** [‡]

[*] *Department of Applied Mathematics and Statistics at Stony Brook University (wesley.suttle@stonybrook.edu).*
[**] *Department of Operations Research and Financial Engineering at Princeton University (zy6@princeton.edu)*
[***] *Coordinated Science Laboratory at University of Illinois at Urbana-Champaign (kzhang66@illinois.edu)*
[****] *Department of Industrial Engineering and Management Sciences at Northwestern University (zhaoran.wang@northwestern.edu).*
[†] *Coordinated Science Laboratory at University of Illinois at Urbana-Champaign (basar1@illinois.edu)*
[‡] *Department of Electrical and Computer Engineering at Stony Brook University (ji.liu@stonybrook.edu).*

Abstract:
This paper extends off-policy reinforcement learning to the multi-agent case in which a set of networked agents communicating with their neighbors according to a time-varying graph collaboratively evaluates and improves a target policy while following a distinct behavior policy. To this end, the paper develops a multi-agent version of emphatic temporal difference learning for off-policy policy evaluation, and proves convergence under linear function approximation. The paper then leverages this result, in conjunction with a novel multi-agent off-policy policy gradient theorem and recent work in both multi-agent on-policy and single-agent off-policy actor-critic methods, to develop and give convergence guarantees for a new multi-agent off-policy actor-critic algorithm. An empirical validation of these theoretical results is given.

Keywords: consensus and reinforcement learning control, adaptive control of multi-agent systems

1. INTRODUCTION

The field of multi-agent reinforcement learning (MARL) has recently seen a flurry of interest in the control and machine learning communities. In this paper, we consider the distributed MARL setting, where a set of agents communicating via a connected but possibly time-varying communication network collaboratively perform policy improvement while sharing only local information. Important recent theoretical works in this area include Kar et al. [2013], where the communication network is incorporated into the underlying model, Zhang et al. [2018b,a], where the theoretical basis for distributed on-policy actor-critic methods is established, Chen et al. [2018], Lin et al. [2019], where progress is made in developing communication-efficient algorithms for this setting, and Doan et al. [2019], where key finite-time results for the multi-agent case are obtained. However, in order for methods based on these recent developments to find widespread future use in important potential application areas – e.g. multi-player games, multi-robot motion planning, and distributed control of energy networks – the development of theoretical tools enabling principled design of data- and resource-efficient algorithms is essential.

Off-policy reinforcement learning with importance sampling correction is an active research area that has recently been leveraged to develop data- and resource-efficient reinforcement learning algorithms for off-policy control. In such methods, an agent seeks to evaluate or improve a given target policy by generating experience according to a distinct behavior policy and reweighting the samples generated to correct for off-policy errors. Algorithms incorporating these methods include Retrace(λ) (Munos et al. [2016], Espeholt et al. [2018]), which use importance sampling to enable reuse of past experience and more intelligent use of multi-processing and parallel computing capabilities. The theory underlying importance sampling-based off-policy methods is relatively well-developed. An off-policy extension of the well-known temporal differences (TD(λ)) algorithm for policy evaluation (Sutton [1995]), called the method of emphatic temporal differences or ETD(λ), has been developed and shown to converge under linear function approximation (Yu [2015], Sutton et al. [2016]). Following the foundational policy gradient theorem of Sutton et al. [2000] for the on-policy case, recent efforts in the area of off-policy policy improvement include Gu et al. [2017], as well as Maei [2018], which builds off the off-policy policy gradient theorem of Degris et al. [2012] in the tabular case to prove convergence of the actor step under linear approximation architectures. Building on the off-policy policy evaluation results in Yu [2015] and Sutton et al. [2016], Imani et al. [2018] provides an off-policy policy gradient theorem using the emphatic weightings that are central to ETD(λ), and describes an off-policy actor-critic algorithm based on their result.

[⋆] All proofs have been omitted due to space considerations and can be found in Suttle et al. [2019]. This research was supported in part by ONR MURI Grant N00014-16-1-2710, in part by the US Army Research Laboratory (ARL) Cooperative Agreement W911NF-17-2-0196, and in part by the Australian Research Council under grants DP-130103610 and DP-160104500, and Data61-CSIRO.

Given the usefulness of off-policy methods in the development of data- and resource-efficient algorithms for reinforcement learning, it is clear that extending such methods to distributed MARL is essential, since such methods can be used to help mitigate slower convergence rates inherited from the distributed setting. In this paper, we present a new off-policy actor-critic algorithm for distributed MARL and provide convergence guarantees. Our algorithm uses a novel multi-agent consensus-based version of ETD(λ) for the critic updates and relies on a new multi-agent off-policy policy gradient theorem using emphatic weightings to enable each agent to compute its portion of the policy gradient for the actor updates. The reader can find empirical results validating our theoretical guarantees in Suttle et al. [2019]. Though the area of off-policy actor-critic methods for distributed MARL is new, and the results we provide in this paper are novel, a very recent work in off-policy actor-critic MARL (Zhang and Zavlanos [2019]) based on gradient temporal differencing (see Lagoudakis and Parr [2003]) appeared within a few days of the first version of our current paper becoming available. We leave comparison of our ETD-based approach and the GTD-based approach in Zhang and Zavlanos [2019] as an important future direction.

2. MODEL FORMULATION

The multi-agent reinforcement learning problem is formulated as a Markov decision process (MDP) model on a time-varying communication network. Let $\mathcal{N} = \{1, \dots, n\}$ denote a set of n agents, and let $\{G_t\}_{t\in\mathbb{N}} = \{(\mathcal{N}, \mathcal{E}_t)\}_{t\in\mathbb{N}}$ denote a possibly time-varying sequence of directed graphs on $\mathcal{N}$, which depicts the neighbor relationships among the agents. Specifically, (j, i) is an edge in G_t whenever agents j and i can communicate. Then, $(S, A, P, \{r^i\}_{i\in\mathcal{N}}, \{G_t\}_{t\in\mathbb{N}}, \gamma)$ characterizes a networked multi-agent discounted MDP, where S is the shared state space, $A = \prod_{i\in\mathcal{N}} A^i$ is the joint action space (which is assumed to be constant, and where A^i is the action space of agent i), $P : S\times S\times A \to [0, 1]$ is the transition probability function, $r^i : S \times A \to [0, 1]$ is the local reward function for each agent $i \in \mathcal{N}$, the sequence $\{G_t\}_{t\in\mathbb{N}}$ describes the communication network at each timestep, and $\gamma \in (0, 1)$ is an appropriately chosen discount factor.

We assume that the state and action spaces are finite. We also assume that, for each graph G_t, there is an associated, nonnegative, possibly random weight matrix C_t that respects the topology of G_t in that, if $(i, j) \notin \mathcal{E}_t$,, then $[C_t]_{ij} = 0$. Several important assumptions about the sequence $\{C_t\}_{t\in\mathbb{N}}$ will be made explicit in Section 6.1 below. Finally, let $\bar{r}_{t+1}$ denote the global reward generated at time $t + 1$, and let $\bar{r} : S \times A \to \mathbb{R}$ be given by $\bar{r}(s, a) = \frac{1}{n}\sum_{i\in\mathcal{N}} r^i(s, a) = E[\bar{r}_{t+1} \mid s_t = s, a_t = a]$.

Recall that a policy function $\nu : A \times S \to [0, 1]$ leads to a conditional probability distribution $\nu(\cdot|s)$ over A for each element $s \in S$. For a given policy ν, the state-value function is

$$v_\nu(s) = E_{s\sim\nu}\Big[\sum_{k=1}^{\infty} \gamma^{k-1}\bar{r}_{t+k} \mid s_t = s\Big],$$

which satisfies

$$v_\nu(s) = \sum_{a\in A} \nu(a|s) \sum_{s'\in S} P(s'|s, a)[\bar{r}(s, a) + \gamma v_\nu(s')].$$

The action-value function is

$$q_\nu(s, a) = \sum_{s'\in S} P(s'|s, a)(\bar{r}(s, a) + \gamma v_\nu(s')).$$

Let each agent $i \in \mathcal{N}$ be equipped with its own local behavior policy $\mu^i : A^i \times S \to [0, 1]$. For each $i \in \mathcal{N}$, let $\pi^i_{\theta^i} : A^i \times S \to [0, 1]$ be some suitable set of local target policy functions parametrized by $\theta^i \in \Theta^i$, where $\Theta^i \subset \mathbb{R}^{m_i}$ is compact. We further assume that each $\pi^i_{\theta^i}$ is continuously differentiable with respect to θ^i. Set $\theta = [\theta_1^T, \dots, \theta_n^T]^T$. Define

$$\mu = \prod_{i=1}^{n} \mu^i : A \times S \to [0, 1] \text{ and } \pi_\theta = \prod_{i=1}^{n} \pi^i_{\theta^i} : A \times S \to [0, 1].$$

These correspond to the global behavior function and global parametrized target policy function, respectively. Assume that $\mu^i(a^i|s) > 0$ whenever $\pi^i_{\theta^i}(a^i|s) > 0$, for all $i \in \mathcal{N}$, all $(a^i, s) \in A^i \times S$, and all $\theta^i \in \Theta^i$. For all $\theta \in \Theta$, assume that the Markov chains generated by π_θ and μ are irreducible and aperiodic, and let $\mathbf{d}_{\pi_\theta}, \mathbf{d}_\mu \in [0, 1]^{|S|}$ denote their respective steady-state distributions, i.e. $d_{\pi_\theta}(s)$ is the steady-state probability of the π_θ-induced chain being in state $s \in S$, and similarly for $d_\mu(s)$.

Finally, let each agent be equipped with a state value function estimator $v_{\omega^i} : S \to \mathbb{R}$ parametrized by $\omega^i \in \Omega$, where $\Omega \subset \mathbb{R}^M, M \in \mathbb{N}, M > 0$ is parameter space shared by all agents. This family of functions will be used in the following to maintain a running approximation of the true value function for the current policy. We emphasize that each agent maintains its own local estimate ω^i of the current value function parameters, but that all agents use identical approximation architectures, i.e. $v_{\omega^i} = v_{\omega^j}$ whenever $\omega^i = \omega^j$. In the case of general approximation architectures, it is only required that v_ω be a suitably expressive approximator that is differentiable in ω, such as a neural network. In our convergence analysis, however, we assume the standard linear approximation architecture $v_\omega(s) = \phi(s)^T\omega$, where $\phi(s)$ is the feature vector corresponding to $s \in S$.

3. EMPHATIC TEMPORAL DIFFERENCE LEARNING

Given our use on ETD(λ), it is helpful to summarize the basic form of single-agent ETD(λ) with linear function approximation in this section. We are given a discounted MDP (S, A, P, r, γ), target policy $\pi : A \times S \to [0, 1]$, and behavior policy $\mu : A \times S \to [0, 1]$, with $\pi \neq \mu$. It is assumed that the steady-state distributions $\boldsymbol{d}_\pi, \boldsymbol{d}_\mu$ of π, μ exist, and that the transition probability matrices that they induce are given by P_π, P_μ. The goal is to perform on-line policy evaluation on π while behaving according to μ over the course of a single, infinitely long trajectory. This is accomplished by carrying out TD(λ)-like updates that incorporate importance sampling ratios to reweight the updates sampled from μ to correspond to samples obtained from π. At a given state-action pair (s, a), the corresponding importance sampling ratio is given by $\rho(s, a) = \frac{\pi(a|s)}{\mu(a|s)}$, with the assumption that if

$\pi(a|s) > 0$, then $\mu(a|s) > 0$, and the convention that $\rho(s,a) = 0$ if $\mu(a|s) = \pi(a|s) = 0$.

The work Yu [2015] proves the convergence of ETD(λ) with linear function approximation using rather general forms of discounting, bootstrapping, and a notion of state-dependent "interest". First, instead of a fixed discount rate $\gamma \in (0,1)$, a state-dependent discounting function $\gamma : S \to [0,1]$ is used. Second, a state-dependent bootstrapping parameter $\lambda : S \to [0,1]$ at each step is allowed. Finally, Yu [2015] include an interest function $i : S \to \mathbb{R}_+$ that stipulates the user-specified interest in each state.

Let $\Phi \in \mathbb{R}^{|S|\times k}$ be the matrix whose rows are the feature vectors corresponding to each state in S, and let $\phi(s)$ denote the row corresponding to state s. Given a trajectory $\{(s_t, a_t)\}_{t\in\mathbb{N}}$, let $\phi_t = \phi(s_t), \rho_t = \rho(s_t,a_t), \gamma_t = \gamma(s_t), \lambda_t = \lambda(s_t)$, and $r_t = r(s_t,a_t)$. An iteration of the general form of ETD(λ) using linear function approximation is as follows:

$$\omega_{t+1} = \omega_t + \alpha_t \rho_t e_t (r_{t+1} + \gamma_{t+1}\phi_{t+1}^T \omega_t - \phi_t^T \omega_t),$$

where the eligibility trace e_t is defined by

$$e_t = \lambda_t \gamma_t \rho_{t-1} e_{t-1} + M_t \phi_t,$$

and M_t is the emphatic weighting given by

$$M_t = \lambda_t i(s_t) + (1-\lambda_t)F_t, \quad F_t = \gamma_t \rho_{t-1} F_{t-1} + i(s_t).$$

The stepsizes $\{\alpha_t\}_{t\in\mathbb{N}}$ satisfy the standard conditions $\alpha_t \geq 0, \sum_t \alpha_t = \infty, \sum_t \alpha_t^2 < \infty$, and (e_0, F_0, ω_0) are specified initial conditions, which may be arbitrary. We refer the reader to Sutton et al. [2016] for an intuitive description and complete derivation of ETD(λ). It is important for our purposes, however, to recognize the projected Bellman equation that it almost surely (a.s.) solves, as well as the associated ordinary differential equation (ODE) that it asymptotically tracks a.s.

Following Yu [2015], let $S = \{s_1, \ldots, s_k\}$ be an enumeration of S. Define diagonal matrices $\Gamma = \text{diag}(\gamma(s_1), \ldots, \gamma(s_k))$ and $\Lambda = \text{diag}(\lambda(s_1), \ldots, \lambda(s_k))$. Let $r_\pi \in \mathbb{R}^k$ be such that its j-th entry is given by $r(s_j, \pi(s_j))$, and define

$$P_{\pi,\gamma}^\lambda = I - (I - P_\pi \Gamma \Lambda)^{-1}(I - P_\pi \Gamma),$$
$$r_{\pi,\gamma}^\lambda = (I - P_\pi \Gamma \Lambda)^{-1} r_\pi.$$

Associated with ETD(λ) is the generalized Bellman equation Sutton [1995], Yu [2015]

$$v = r_{\pi,\gamma}^\lambda + P_{\pi,\gamma}^\lambda v,$$

with unique solution which we denote by v_π. ETD(λ) solves the projected Bellman equation

$$v = \Pi(r_{\pi,\gamma}^\lambda + P_{\pi,\gamma}^\lambda v), \tag{1}$$

where v is constrained to lie in the column space of Φ, and Π is the projection onto colsp(Φ) with respect to the Euclidean norm weighted by the diagonal matrix

$$\overline{M} = \text{diag}(\boldsymbol{d}_{\mu,i}^T (I - P_{\pi,\gamma}^\lambda)^{-1}),$$

where $\boldsymbol{d}_{\mu,i}(s_j) = \boldsymbol{d}_\mu(s_j) \cdot i(s_j)$, for $j = 1, \ldots, k$. It does this by finding the solution to the equation

$$D\omega + b = 0, \tag{2}$$

where $\omega \in \mathbb{R}^k$ is the element in the approximation space $\mathbb{R}^k$ corresponding to the linear combination $\Phi\omega \in$ colsp(Φ), and D and b are given by

$$D = -\Phi^T \overline{M}(I - P_{\pi,\gamma}^\lambda)\Phi, \quad b = \Phi^T \overline{M} r_{\pi,\gamma}^\lambda.$$

When D is negative definite, ETD(λ) is proven in Yu [2015] to almost surely find the unique solution $\omega^* = -D^{-1}b$ of equation (2) above, which is equivalent to finding the unique element $\Phi\omega^* \in$ colsp(Φ) solving (1).

In our extension of ETD(λ) to the multi-agent case, we make the notation-simplifying assumptions that $\gamma(s) = \gamma \in (0,1)$ and $\lambda(s) = \lambda \in [0,1]$, and $i(s) = 1$, for all $s \in S$.

4. MULTI-AGENT OFF-POLICY POLICY GRADIENT THEOREM

Following Degris et al. [2012] and Imani et al. [2018], when performing gradient ascent on the global policy function, we seek to maximize

$$J_\mu(\theta) = \sum_{s\in S} d_\mu(s) v_{\pi_\theta}(s). \tag{3}$$

For an agent to perform its gradient update at each actor step, it needs access to an estimate of its portion of the policy gradient. In the single-agent case, Imani et al. [2018] obtains the expression

$$\nabla_\theta J_\mu(\theta) = \sum_{s\in S} m(s) \sum_{a\in A} [\nabla_\theta \pi_\theta(a|s)] q_{\pi_\theta}(s,a),$$

for the policy gradient, where $m(s)$ is the emphatic weighting of $s \in S$, with vector form $\mathbf{m}^T = \mathbf{d}_\mu^T (\mathbf{I} - P_{\theta,\gamma})^{-1}$, where $P_{\theta,\gamma} \in \mathbb{R}^{|S|\times|S|}$ has entries given by

$$P_{\theta,\gamma}(s,s') = \gamma \sum_{a\in A} \pi_\theta(a|s) P(s'|s,a).$$

Recall that θ^i is the parameter of the local target policy $\pi_{\theta^i}^i$, $\forall i \in \mathcal{N}$. We will henceforth use the shorthand q_θ to refer to the action-value function q_{π_θ} of policy π_θ. Building on the work in Imani et al. [2018] and Zhang et al. [2018b], which themselves are built on Sutton et al. [2000], for the multi-agent case we obtain the following expression for the off-policy policy gradient in the multi-agent case, the proof of which can be found in Suttle et al. [2019]:

Theorem 4.1. *The gradient of $J_\mu(\theta)$ defined in (3) with respect to each θ^i is*

$$\nabla_{\theta^i} J_\mu(\theta) = \sum_{s\in S} m(s) \sum_{a\in A} \pi_\theta(a|s) q_\theta(s,a) \nabla_{\theta^i} \log \pi_{\theta^i}(a^i|s). \tag{4}$$

It is also possible to incorporate baselines similar to those in Zhang et al. [2018b] in this expression, and the derivations are similar to those in that paper.

Let ρ_t, F_t be as in the previous section, and let $\delta_t^i = r_{t+1}^i + \gamma v_{\omega_t^i}(s_{t+1}) - v_{\omega_t^i}(s_t)$ denote the temporal difference of the actor update at agent i at time t. For the actor portion of our algorithm, we need a slightly different emphatic weighting update than that in ETD(λ), corresponding to the update used in Imani et al. [2018]. Define

$$M_t^\theta = (1 - \lambda_t^\theta) + \lambda^\theta F_t = 1 + \lambda^\theta \gamma \rho_{t-1} F_{t-1}.$$

In the actor portion of our algorithm given in the next section, we will be sampling from the expectation

$$E_\mu[\rho_t M_t^\theta \delta_t^i \nabla_{\theta^i} \log \pi_{\theta_t^i}^i(a_t|s_t)] \tag{5}$$

and using it as an estimate of the policy gradient at each timestep. To see why sampling from (5) might give us an estimate of the desired gradient, note that, for fixed θ,

$$\sum_{a\in A} \pi_\theta(a|s) q_\theta(s,a) \nabla_{\theta^i} \log \pi_{\theta^i}^i(a^i|s)$$

$$= \sum_{a\in A} \mu(a|s)\rho_\theta(s,a)q_\theta(s,a)\nabla_{\theta^i}\log\pi^i_{\theta^i}(a^i|s).$$

To justify this sampling procedure, it is also important to note that, given the true q_{θ_t} for policy π_{θ_t}, such sampling leads to unbiased estimates, i.e.

$$\sum_{s\in S} m(s)\sum_{a\in A} q_{\theta_t}(s_t,a_t)\nabla_{\theta^i}\pi_{\theta^i_t}(a^i_t|s_t)$$
$$= E_\mu[\rho_t M^\theta_t \delta^i_t \nabla_{\theta^i}\log\pi^i_{\theta^i_t}(a_t|s_t)]. \quad (6)$$

Proof of (6) in the single-agent case can be found in Imani et al. [2018], and the multi-agent case is an immediate consequence.

5. ALGORITHMS

5.1 Single-agent Algorithm

Before introducing our multi-agent algorithm, we first describe the single-agent version. This is a two-timescale off-policy actor-critic algorithm, where the critic updates are carried out at the faster timescale using ETD(λ), while the actor updates are performed at the slower timescale using the emphatically-weighted updates as in the previous section. The form of the following algorithm is based on Imani et al. [2018], but we choose an explicit method for performing the ω updates.

Let $\omega \in \Omega \subset \mathbb{R}^k$ and $\theta \in \Theta \subset \mathbb{R}^l$ be the value function and policy function parameters, respectively. For now, we can simply take $\Omega = \mathbb{R}^k$ and $\Theta = \mathbb{R}^l$. We will impose conditions on them (Θ, in particular) in the Assumptions section below. First initialize the parameters by setting $\theta_0 = 0,\ \omega_0 = e_{-1} = 0,\ F_{-1} = 0,\ \rho_{-1} = 1$.[1] In each iteration, execute action $a_t \sim \mu(\cdot|s_t)$ and observe r_{t+1} and s_{t+1}, then update the emphatic weightings by

$$M_t = \lambda + (1-\lambda)F_t, \qquad M^\theta_t = 1 + \lambda^\theta \gamma\rho_{t-1}F_{t-1},$$

with $F_t = 1 + \gamma\rho_{t-1}F_{t-1}$. Finally, update the actor and critic parameters using the emphatic weightings:

$$\omega_{t+1} = \omega_t + \beta_{\omega,t}\rho_t(r_{t+1} + \gamma v_{\omega_t}(s_{t+1}) - v_{\omega_t}(s_t))e_t,$$
$$\theta_{t+1} = \theta_t + \beta_{\theta,t}\rho_t M^\theta_t \nabla_\theta \log\pi_{\theta_t}(a_t|s_t)\delta_t,$$

where e_t is given by $e_t = \gamma\lambda e_{t-1} + M_t\nabla_\omega v_{\omega_t}(s_t)$, and $\delta_t = r_{t+1} + \gamma v_{\omega_t}(s_{t+1}) - v_{\omega_t}(s_t)$ is the standard TD(0) error. It is important to mention that δ_t can also be regarded as an estimate of the advantage function $q_\pi(s_t,a_t) - v_\pi(s_t)$, which is the standard example of including baselines.

5.2 Multi-agent Algorithm

The overall structure of the multi-agent algorithm is similar to the single-agent version, with two key differences: (i) the agents perform the critic updates at the faster timescale using one consensus process to average their current ω estimates and an inner consensus process to obtain the importance sampling ratios necessary to perform ETD(λ) for the current "static" global policy; (ii) each agent is responsible for updating only its own portion of the policy gradient at each actor update at the slower timescale.

[1] Imani et al. [2018] suggests $\lambda^\theta = 0.9$ as a default value. We currently have no suggestions for λ.

All agents are initialized as in the single-agent case. At each step, each agent first performs a consensus average of its neighbor's ω-estimates, selects its next action, and computes its local importance sampling ratio. Specifically, at the t-th iteration, agent i first receives $\widetilde{\omega}^j_{t-1}$ from each of its neighbors $j \in \mathcal{N}_t(i)$, executes its own action $a^i_t \sim \mu_i(\cdot|s_t)$, and observes the joint action a_t, its own reward r^i_{t+1}, and the next state s_{t+1}. Agent i then aggregates the information obtained from its neighbors with the consensus update $\omega^i_t = \sum_{j\in\mathcal{N}} c_{t-1}(i,j)\widetilde{\omega}^j_{t-1}$, and also computes the log of its local importance sampling ratio

$$p^i_t = \log\big[\pi^i_{\theta^i_t}(a^i_t|s_t)/\mu_i(a^i_t|s_t)\big].$$

Here $c_t(i,j)$ is the communication weight from agents j to i at time t. For undirected graphs, one particular choice of the weights $c_t(i,j)$ that relies on only local information of the agents is known as the Metropolis weights (Xiao et al. [2005]) given by

$$c_t(i,j) = \big(1 + \max[d_t(i), d_t(j)]\big)^{-1}, \quad \forall (i,j) \in \mathcal{E}_t,$$
$$c_t(i,i) = 1 - \sum_{j\in\mathcal{N}_t(i)} c_t(i,j), \quad \forall i \in \mathcal{N},$$

where $\mathcal{N}_t(i) = \{j \in \mathcal{N} \colon (j,i) \in \mathcal{E}_t\}$ is the set of neighbors of agent i at time t, and $d_t(i) = |\mathcal{N}_t(i)|$ is the number of neighbors at time t.

Next, the agents enter an inner loop and perform the following, repeating until a consensus average of the original values is achieved. In each iteration of the inner loop, each agent i broadcasts its local p^i_t to its neighbors and receives p^j_t from each neighbor $j \in \mathcal{N}_t(i)$. Agent i then updates its local log importance sampling ratio via $p^i_t \leftarrow \sum_{j\in\mathcal{N}} c_t(i,j)p^j_t$. Such an iteration is repeated until consensus is reached, and all the agents break out of the inner loop. For directed graphs, the average consensus can be achieved by using the idea of the push-sum protocol Kempe et al. [2003]; see Liu and Morse [2012] for a detailed description of the algorithm. After achieving consensus, $p^i_t = p^j_t$ for all $i,j \in \mathcal{N}$. Notice that $p^i_t = \frac{1}{n}\sum_{i=1}^n \log\rho^i_t$, so that $\exp(np^i_t) = \exp(\sum_{i=1}^n \log\rho^i_t) = \prod_{i=1}^n \rho^i_t = \rho_t = \pi_{\theta_t}(a_t|s_t)/\mu(a_t|s_t)$.

Each agent then performs the local critic and actor updates. For the critic update, agent i first computes the emphatic weighting and the importance sampling ratio

$$M_t = \lambda + (1-\lambda)F_t, \qquad \rho_t = \exp(np^i_t),$$

where F_t is given by $F_t = 1 + \gamma\rho_{t-1}F_{t-1}$. Notice that this update will be identical across agents. Then, agent i updates its critic parameter $\widetilde{\omega}^i_t$ via

$$e_t = \gamma\lambda e_{t-1} + M_t\nabla_\omega v_{\omega^i_t}(s_t),$$
$$\widetilde{\omega}^i_t = \omega^i_t + \beta_{\omega,t}\rho_t\delta^i_t e_t,$$

where $\delta^i_t = r^i_{t+1} + \gamma v_{\omega^i_t}(s_{t+1}) - v_{\omega^i_t}(s_t)$ is the TD-error computed locally by agent i. The parameter $\widetilde{\omega}^i_t$ is then broadcast to all the neighbors in $\mathcal{N}_t(i)$. Finally, for the actor update, the emphatic weighting M^θ_t is obtained by

$$M^\theta_t = 1 + \lambda^\theta\gamma\rho_{t-1}F_{t-1},$$

and the parameter of the local policy π^i_θ is updated via

$$\theta^i_{t+1} = \theta^i_t + \beta_{\theta,t}\rho_t M^\theta_t \nabla_{\theta^i}\log\pi^i_{\theta^i_t}(s_t,a^i_t)\delta^i_t.$$

A concise presentation of the algorithm is given below.

Algorithm 1 Multi-agent Off-policy Actor-critic

Initialize $\theta_0^i = 0, \omega_0 = e_{-1} = 0, F_{-1} = 0, \rho_{-1} = 1$, for all $i \in \mathcal{N}$, the initial state s_0, and the stepsizes $\{\beta_{\omega,t}\}_{t\in\mathbb{N}}, \{\beta_{\theta,t}\}_{t\in\mathbb{N}}$.

repeat
  for all $i \in \mathcal{N}$ **do**
    receive $\widetilde{\omega}_{t-1}^j$ from neighbors $j \in \mathcal{N}_t(i)$
    $\omega_t^i = \sum_{j\in\mathcal{N}} c_{t-1}(i,j)\widetilde{\omega}_{t-1}^j$
    execute $a_t^i \sim \mu_i(\cdot|s_t)$
    $\rho_t^i = \frac{\pi_{\theta_t^i}^i(a_t^i|s_t)}{\mu_i(a_t^i|s_t)}$
    $p_t^i = \log \rho_t^i$
    observe r_{t+1}^i, s_{t+1}
    repeat
      broadcast p_t^i, receive p_t^j from $j \in \mathcal{N}_t(i)$
      $p_t^i \leftarrow \sum_{j\in\mathcal{N}} c_t(i,j)p_t^j$
    until consensus is achieved
    $\rho_t = \exp(np_t^i)$
    $F_t = 1 + \gamma\rho_{t-1}F_{t-1}$
    $M_t = \lambda + (1-\lambda)F_t$
    $e_t = \gamma\lambda e_{t-1} + M_t\nabla_\omega v_{\omega_t^i}(s_t)$
    $\delta_t^i = r_{t+1}^i + \gamma v_{\omega_t^i}(s_{t+1}) - v_{\omega_t^i}(s_t)$
    $\widetilde{\omega}_t^i = \omega_t^i + \beta_{\omega,t}\rho_t\delta_t^i e_t$
    $M_t^\theta = 1 + \lambda^\theta\gamma\rho_{t-1}F_{t-1}$
    $\theta_{t+1}^i = \theta_t^i + \beta_{\theta,t}\rho_t M_t^\theta \nabla_{\theta^i}\log\pi_{\theta_t^i}^i(a_t^i|s_t)\delta_t^i$
    broadcast $\widetilde{\omega}_t^i$ to neighbors over network
  end for
until convergence

6. THEORETICAL RESULTS

As is standard in two-timescale stochastic approximation schemes Borkar [2009], in our convergence analysis we first prove the a.s. convergence of the faster timescale updates while viewing the slower timescale θ and corresponding policy π_θ as static, and then show the a.s. convergence of the θ updates while viewing the value of the faster timescale ω as equilibrated at every timestep. In our case, we have the additional complications that experience is being generated by each agent i according to a fixed behavior policy μ^i, the critic updates are achieved using a multi-agent, consensus-based version of ETD(λ), and we are using an off-policy gradient sampling scheme in our actor updates, but our convergence analysis still follows this two-stage pattern: Theorem 6.1 provides convergence of the critic updates, while Theorem 6.2 provides convergence of the actor updates. See Suttle et al. [2019] for proofs as well as an empirical evaluation of our theoretical results.

6.1 Assumptions

Assumptions 6.1.1, 6.1.2, and 6.1.3 are standard conditions taken from Zhang et al. [2018b]. 6.1.4 is a standard condition in stochastic approximation. 6.1.5 requires that the behavior policy be sufficiently exploratory, and also allows us to bound the importance sampling ratios ρ_t, which is critical in our convergence proofs. 6.1.6 simplifies the convergence analysis in the present work, but, as mentioned above, the assumption can likely be weakened or removed by carefully bounding the errors resulting from terminating the inner loop after a specified level of precision is achieved.

Assumption 6.1.1. *For each agent $i \in \mathcal{N}$, the local θ-update is carried out using the projection operator $\Gamma^i : \mathbb{R}^{m_i} \to \Theta^i \subset \mathbb{R}^{m_i}$. Furthermore, the set $\Theta = \prod_{i=1}^n \Theta^i$ contains at least one local optimum of $J_\mu(\theta)$.*

Assumption 6.1.2. *For each element $C_t \in \{C_t\}_{t\in\mathbb{N}}$,*

(1) C_t is row stochastic, $E[C_t]$ is column stochastic, and there exists $\alpha \in (0,1)$ such that, for any $c_t(i,j) > 0$, we have $c_t(i,j) \geq \alpha$.
(2) If $(i,j) \notin \mathcal{E}_t$, we have $c_t(i,j) = 0$.
(3) The spectral norm $\rho = \rho(E[C_t^T(I - \mathbf{1}\mathbf{1}^T/N)C_t])$ satisfies $\rho < 1$.
(4) Given the σ-algebra $\sigma(C_\tau, \{r_\tau^i\}_{i\in\mathcal{N}}; \tau \leq t)$, C_t is conditionally independent of r_{t+1}^i for each $i \in \mathcal{N}$.

Assumption 6.1.3. *The feature matrix Φ has linearly independent columns, and the value function approximator $v_\omega(s) = \phi(s)^T\omega$ is linear in ω.*

Assumption 6.1.4. *$\sum_t \beta_{\omega,t} = \sum_t \beta_{\theta,t} = \infty$, $\sum_t \beta_{\omega,t}^2 + \beta_{\theta,t}^2 < \infty$, $\beta_{\theta,t} = o(\beta_{\omega,t})$, and $\lim_{t\to\infty} \frac{\beta_{\omega,t+1}}{\beta_{\omega,t}} = 1$.*

Assumption 6.1.5. *For some fixed $0 < \varepsilon \leq \frac{1}{|S|\cdot|A|}$, we have $\varepsilon \leq \mu(a|s)$, for all state-action pairs $(s,a) \in S \times A$.*

Assumption 6.1.6. *Each agent performs its update at timestep t using the exact value of ρ_t.*

6.2 Convergence

For the first step of our analysis we prove that, for a fixed target policy π_θ and behavior policy μ, when using linear function approximation the multi-agent version of ETD(λ) given in the critic step of our algorithm converges in the following sense: almost surely, each agent asymptotically obtains a copy of the unique solution $\omega_\theta \overset{\text{def}}{=} \omega^* = -D^{-1}b$ described in Section 3, which provides each agent with the best approximator $\Phi\omega_\theta$ of the global value function v_θ for the multi-agent MDP under policy π_θ. More concisely:

Theorem 6.1. *Given a fixed target policy π_θ and behavior policy μ, multi-agent ETD(λ) achieves consensus a.s. when using linear function approximation, and, under Assumption 6.1.3, the consensus vector is a.s. the unique solution of (2).*

For the second step of our analysis, we show that the vector $\theta_t = \left[(\theta_t^1)^T \ \dots \ (\theta_t^n)^T\right]^T$ of the agents' policy parameters converges a.s. to an equilibrium point θ^* of a certain ODE ((7) given below). Let

$$A_t^i = r_{t+1}^i + \gamma\phi_{t+1}^T\omega_t^i - \phi_t^T\omega_t^i, \quad \psi_t^i = \nabla_{\theta^i}\log\pi_{\theta_t^i}^i(a_t|s_t),$$

and $\mathcal{G}_t = \sigma(\theta_\tau; \tau \leq t)$ be the σ-algebra generated by the θ-iterates up to time t. Define

$$A_{t,\theta}^i = r_{t+1}^i + \gamma\phi_{t+1}^T\omega_\theta - \phi_t^T\omega_\theta,$$

where ω_θ is the limit of the critic step at the faster timestep under target policy π_θ. We then have the following:

Theorem 6.2. *The update*

$$\theta_{t+1}^i = \Gamma^i(\theta_t^i + \beta_{\theta,t}\rho_t M_t^\theta A_t^i\psi_t^i) \tag{7}$$

converges a.s. to the set of asymptotically stable equilibria of the ODE

$$\dot{\theta}^i = \hat{\Gamma}^i(h^i(\theta)), \tag{8}$$

where $h^i(\theta_t) = E[\rho_t M_t^\theta A^i_{t,\theta_t} \psi^i_t \mid \mathcal{G}_t]$ *and* $\hat{\Gamma}(h(x)) = \lim_{\epsilon \downarrow 0} \frac{\Gamma(x+\epsilon h(x))-x}{\epsilon}$.

Theorem 6.2 is in the same vein as the classic convergence results for single-agent actor-critic under linear function approximation architectures Bhatnagar et al. [2009], Bhatnagar [2010], Degris et al. [2012]. For discussion of the definition of $\hat{\Gamma}$, see the section on the Kushner-Clark lemma in Suttle et al. [2019]. It is important to note that, since the approximation $\Phi\omega_\theta$ obtained during the critic step is in general a biased estimate of the true value function v_θ, the term $\rho_t M_t^\theta A^i_{t,\theta} \phi^i_t$ will usually also be a biased estimate of the true policy gradient. However, given that the error between the true value function v_{θ^*} and the estimate $\Phi\omega_{\theta^*}$ is small, the point θ^* will lie within a small neighborhood of a local optimum of (3), as noted in Zhang et al. [2018b].

7. CONCLUSIONS

In this paper we have rigorously extended off-policy actor-critic methods to the multi-agent reinforcement learning context. Based on these foundations, promising future directions include exploring additional theoretical applications of multi-agent emphatic temporal difference learning, practical and theoretical methods for handling Assumption 6.1.6, empirical comparison of our algorithm with other off-policy multi-agent reinforcement learning algorithms, and the development of practical applications.

REFERENCES

Bhatnagar, S. (2010). An actor–critic algorithm with function approximation for discounted cost constrained markov decision processes. *Systems & Control Letters*, 59(12), 760–766.

Bhatnagar, S., Sutton, R.S., Ghavamzadeh, M., and Lee, M. (2009). Natural actor–critic algorithms. *Automatica*, 45(11), 2471–2482.

Borkar, V.S. (2009). *Stochastic approximation: a dynamical systems viewpoint*. Springer.

Chen, T., Zhang, K., Giannakis, G.B., and Başar, T. (2018). Communication-efficient distributed reinforcement learning. *arXiv preprint arXiv:1812.03239*.

Degris, T., White, M., and Sutton, R.S. (2012). Off-policy actor-critic. In *29th International Conference on Machine Learning*, 179–186.

Doan, T., Maguluri, S., and Romberg, J. (2019). Finite-time analysis of distributed TD(0) with linear function approximation on multi-agent reinforcement learning. In *36th International Conference on Machine Learning*, 1626–1635.

Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., Doron, Y., Firoiu, V., Harley, T., Dunning, I., et al. (2018). Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *35th International Conference on Machine Learning*, 1407–1416.

Gu, S., Lillicrap, T., Ghahramani, Z., Turner, R.E., and Levine, S. (2017). Q-prop: Sample-efficient policy gradient with an off-policy critic. In *International Conference on Learning Representations*.

Imani, E., Graves, E., and White, M. (2018). An off-policy policy gradient theorem using emphatic weightings. In *Advances in Neural Information Processing Systems*, 96–106.

Kar, S., Moura, J.M., and Poor, H.V. (2013). QD-learning: A collaborative distributed strategy for multi-agent reinforcement learning through consensus + innovations. *IEEE Transactions on Signal Processing*, 61(7), 1848–1862.

Kempe, D., Dobra, A., and Gehrke, J. (2003). Gossip-based computation of aggregate information. In *44th Annual IEEE Symposium on Foundations of Computer Science*, 482–491.

Lagoudakis, M.G. and Parr, R. (2003). Least-squares policy iteration. *Journal of machine learning research*, 4(Dec), 1107–1149.

Lin, Y., Zhang, K., Yang, Z., Wang, Z., Başar, T., Sandhu, R., and Liu, J. (2019). A communication-efficient multi-agent actor-critic algorithm for distributed reinforcement learning. In *58th Conference on Decision and Control*, 5562–5567.

Liu, J. and Morse, A.S. (2012). Asynchronous distributed averaging using double linear iterations. In *2012 American Control Conference*, 6620–6625.

Maei, H.R. (2018). Convergent actor-critic algorithms under off-policy training and function approximation. *arXiv preprint arXiv:1802.07842*.

Munos, R., Stepleton, T., Harutyunyan, A., and Bellemare, M. (2016). Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems*, 1054–1062.

Suttle, W., Yang, Z., Zhang, K., Wang, Z., Başar, T., and Liu, J. (2019). A multi-agent off-policy actor-critic algorithm for distributed reinforcement learning. *arXiv preprint arXiv:1903.06372*.

Sutton, R.S. (1995). TD models: Modeling the world at a mixture of time scales. In *Machine Learning Proceedings 1995*, 531–539. Elsevier.

Sutton, R.S., Mahmood, A.R., and White, M. (2016). An emphatic approach to the problem of off-policy temporal-difference learning. *The Journal of Machine Learning Research*, 17(1), 2603–2631.

Sutton, R.S., McAllester, D.A., Singh, S.P., and Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, 1057–1063.

Xiao, L., Boyd, S., and Lall, S. (2005). A scheme for robust distributed sensor fusion based on average consensus. In *Fourth International Symposium on Information Processing in Sensor Networks*, 63–70.

Yu, H. (2015). On convergence of emphatic temporal-difference learning. In *Conference on Learning Theory*, 1724–1751.

Zhang, K., Yang, Z., and Başar, T. (2018a). Networked multi-agent reinforcement learning in continuous spaces. In *57th Conference on Decision and Control*, 2771–2776.

Zhang, K., Yang, Z., Liu, H., Zhang, T., and Başar, T. (2018b). Fully decentralized multi-agent reinforcement learning with networked agents. In *35th International Conference on Machine Learning*, 9340–9371.

Zhang, Y. and Zavlanos, M.M. (2019). Distributed off-policy actor-critic reinforcement learning with policy consensus. *arXiv preprint arXiv:1903.09255*.