Fully Distributed Policy Evaluation with Local Agent Updates over Time-Varying Communication Network

Matthew Crespo
Electrical and Computer Engineering
University of Central Florida
Orlando, United States of America
matthew.crespo@ucf.edu

Chinwendu Enyioha
Electrical and Computer Engineering
University of Central Florida
Orlando, United States of America
cenyioha@ucf.edu

Abstract

This paper studies a consensus-based policy evaluation algorithm in a cooperative team of heterogeneous learners. To improve each agent's approximation of their value function, they each update their weight parameters, then perform consensus updates with neighboring agents over a dynamic communication network. We present the analysis of an efficient fully distributed algorithm for cooperatively evaluating and optimizing their value functions over a *time-varying* communication network in the average-reward setting. Our main result shows that, using this algorithm, the agents can approximate the true value function, with the approximation error dependent on the number of communication rounds and number of time steps it takes for the communication network to be connected. To validate the theoretical results, we present accompanying numerical experiments that show the theoretical error bounds are not only tight, but get tighter as the time steps required to guarantee network connectivity reduces.

Index Terms

MARL, time-varying communication, fully distributed, policy evaluation, TD learning

I. Introduction

This work studies the problem of multi-agent reinforcement learning (MARL) in which a joint action taken by all agents affects the shared environment. Specifically, we focus on policy evaluation (PE), in which each agent interacts with the environment and shares its weight parameters with neighboring agents (performs a consensus update) over a time-varying communication network to improve their local approximation of the value function. At each time step, all the agents take a joint action which determine each agent's reward and the next state of the entire system. Though rewards obtained at each state are local to each agent, the goal of MARL is for *all* agents to jointly maximize the global average long-term reward of all the agents. In this paper, we study a distributed protocol in which the heterogeneous learners, connected over a dynamic communication network share their weights with one another. Cooperative learning amongst a number of heterogeneous agents is a well-studied problem [1]–[3]. Building on the success of single agent reinforcement learning, MARL has emerged as a tool for solving decision problems in many contexts, including robotics, power and autonomous systems [4], [5]. As the agents collaborate to learn, policy evaluation is a key step as it enables assessment of states for a given policy with the goal of improving the policy. For this process to occur in a multi-agent context, communication is needed amongst the agents.

The main contribution of this work is a convergence analysis on a sample-efficient algorithm for the average-reward multi-agent PE problem over a *dynamic* communication network. In the setting considered, a set of agents cooperatively evaluate the value function of the global states for a given policy via local interactions with neighboring agents over a time-varying network. Not requiring network connectivity at each time step and allowing for links to be broken and established arbitrarily as the agents exchange their weights changes the consensus dynamics as the agents converge to an equilibrium point. This assumption contrasts existing work on MARL where stronger assumptions on inter-agent interactions are made [1], [6].

The rest of the paper is structured as follows. In Section II, we give a brief background and literature review in cooperative learning amongst groups of agents to place our work in context; and follow in Section III with the MARL policy evaluation problem set up in the time-varying communication setting. Our main convergence result is presented in Section IV. Illustrations of our results are presented in Section V, where we also compare them against other results in MARL policy evaluations.

II. Related Work

Recent advancements in MARL have focused on developing algorithms that are aimed on improving collective rewards by adapting value-based methods including approaches like Independent Q-Learning (IQL) [7], where traditional Q-learning is applied to each agent independently to estimate optimal actions based on local observations. While straightforward, this method often struggles with convergence and stability due to the non-stationarity introduced by other learning agents.

On the other hand, policy gradient methods introduced in [8] and further explored by [9], [10], allow agents to directly parameterize their policy and optimize it based on the expected reward. Among the various methods used for policy evaluation, Temporal Difference (TD) learning, introduced in [11], stands out due to its efficiency and practicality in online learning scenarios. Unlike Monte Carlo methods, which require complete episodes to update value estimates, TD learning uses the differences between the estimated value of the current state and the value of the next state, referred to as the TD error. This error signal is used to adjust the value estimates incrementally, enabling agents to learn in real time.

Despite significant progress, MARL faces several challenges. One major issue is the non-stationarity of the environment. This is the basis of the problem addressed in this paper. Researchers in the distributed optimization literature have explored how to maintain convergence in such settings, demonstrating that by appropriately designing communication protocols, agents can achieve optimal solutions even when the network topology changes [12]–[14]. Since TD learning is not really a gradient-based method and state distributions could be different at each time steps due to the underlying Markov process, convergence analysis techniques of gradient-based methods do not easily fit the cooperative MARL problem studied here.

Closely related to this work are [15] where the policy evaluation problems are studied over time-varying communication network in the discounted reward setting with no local updates using the vanilla TD algorithm; and [1] where a similar problem is studied with the assumption that the network is time-invariant. Our focus here is on the consensus-based policy evaluation algorithm using *linear* function approximation, as it is known that TD-learning-based policy evaluation may fail to converge with nonlinear function approximation [16]. The outcomes from this work establishes conditions under which convergence is achieved, and presents tools to understand the impact of network characteristics on the algorithm performance in real-world implementation.

III. MARL Policy Evaluation Problem

In this section we introduce the mathematical preliminaries for our theoretical analysis on the temporal difference algorithm with local TD updates in the case of time-varying communication.

Notation: The l_2-norm of a vector is denoted as $\|\cdot\|$, and the Frobenius norm of matrices as $\|\cdot\|_F$. Furthermore, $\prod_{n=0}^{N} A_n$ is used to denote a product of matrices in which the last matrix in the sequence is multiplied by the previous matrix from the right; for example, $\prod_{n=0}^{1} A_n = A_0 A_1$ in the case when $N = 1$. We consider a time-varying undirected graph defined as $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t)$, comprising a set of nodes $\mathcal{V}$ and edge set $\mathcal{E}_t \subseteq \mathcal{V} \times \mathcal{V}$ at time t. A graphical description of a time-varying graph is presented in Figure 1 over three time steps. Finally, the expectation operator is denoted as $\mathbb{E}[\cdot]$ and the cardinality of a set is $|\cdot|$.

A. Multi-Agent Reinforcement Learning Setting

The model for this work is comprised of N agents connected over a time-varying communication network represented by a graph $\mathcal{G}_t$ at time t. More broadly, we define this multi-agent Markov Decision Process (MDP) with the following five-tuple: $(\mathcal{S}, \{\mathcal{A}^i\}_{i\in\mathcal{V}}, P, \{r^i\}_{i\in\mathcal{V}}, \mathcal{G}_t)$, where $\mathcal{S}$ is the state-space of the entire system, $\{\mathcal{A}^i\}_{i\in\mathcal{V}}$ the collection of action spaces for each agent $i \in \mathcal{V}$. The variable $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ represents the global state transition matrix, $\{r^i\}$ is the reward for agent i given by $r^i : \mathcal{S} \times \mathcal{A}$, and $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t)$ the time-varying graph of the network, where $t = lK$ since the communication rounds will be done every K time steps. For clarity, $\mathcal{A} = \prod_{i\in\mathcal{V}} \mathcal{A}^i$ is the joint action space of the agents which means that the transition matrix is dependent on the entire state of the system at two time steps and the joint action of all the agents in the system.

The goal of MARL in the average reward setting is to maximize the long-term average reward for a given MDP as defined by the 5-tuple above. The long term average reward of a multi-agent system is written mathematically as

$$\begin{aligned} J_\pi &= \lim_{T\to\infty} \frac{1}{T}\mathbb{E}\Big[\sum_{t=0}^{T-1} \frac{1}{N} \sum_{i\in\mathcal{V}} r^i_{t+1}\Big] \\ &= \sum_{s\in\mathcal{S}} d(s) \sum_{a\in\mathcal{A}} \pi(a|s)\bar{r}(s, a), \end{aligned}$$

where $d(\cdot)$ is the steady state distribution of the policy π, and $\bar{r}(s, a)$ is the average reward of all the agents $i \in \mathcal{V}$.

B. Problem

In the case of a large state-space $\mathcal{S}$, the goal of policy evaluation (PE) is to determine the best approximation of the true value function for a given policy π. Given some policy $\pi^i(a|s)$, each agent i updates the weight vector of the state-value function approximation using the update equation

$$w^i_{l,k+1} = w^i_{l,k} + \beta \cdot \delta^i_{l,k} \cdot \phi(s_{l,k}), \tag{1}$$

where $w^i_{l,k+1}$ is agent i's weight at time $k+1$ after l consensus updates/communication rounds, β is the step size, and $\delta^i_{l,k}$ is the TD error after l consensus steps at time step k. Despite the form of Equation (1), TD(0) is not a gradient-based method since

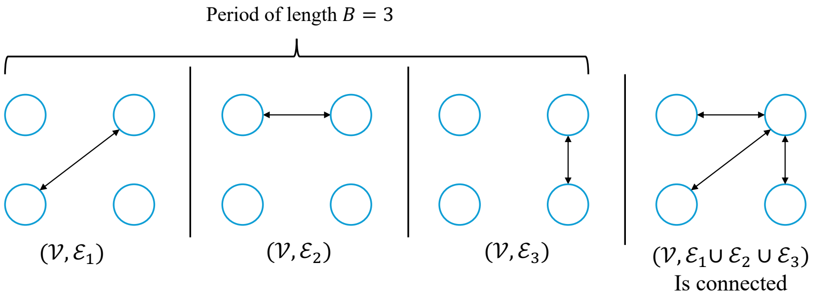

Period of length $B = 3$
$(\mathcal{V}, \mathcal{E}_1)$
$(\mathcal{V}, \mathcal{E}_2)$
$(\mathcal{V}, \mathcal{E}_3)$
$(\mathcal{V}, \mathcal{E}_1 \cup \mathcal{E}_2 \cup \mathcal{E}_3)$
Is connected

Fig. 1: Time-varying Graph in which the edge set at each time changes arbitrarily. Definition 1 presents an upper bound on the time interval over which a sequence of graphs is connected. This figure shows a B-connected graph where $B = 3$.

an objective function is not being directly optimized—this method uses the current value function approximation, exponentially weighted average reward, and most recent reward to determine an update to the weight vector w. The TD error $\delta^i_{l,k}$ given by

$$\delta^i_{l,k} = r^i_{l,k+1} - \mu^i_{l,k} + \phi(s_{l,k+1})^\top w^i_{l,k} - \phi(s_{l,k})^\top w^i_{l,k},$$

is used to determine whether the weight vectors will be updated in the same or opposite direction to feature vector for a given state. In addition, an exponentially weighted average

$$\mu^i_{l,k+1} = (1-\beta)\mu^i_{l,k} + \beta r^i_{l,k+1},$$

is calculated where β is a parameter that more heavily weighs recent rewards as compared to rewards gained earlier in the learning process. After a given number K of local updates represented by Equation (1), each agent i carries out a consensus update with its neighboring set of connected agents, as given by Definition 2, following

$$w^i_{l+1,0} = \sum_{j \in \mathcal{N}_i(t)} A_{ij}(t) w^j_{l,K}, \tag{2}$$

where $A_{ij}(t)$ is the (i,j)th entry of the adjacency matrix $A(t)$ representing the communication graph at time step t. Our focus is on convergence of the preceding updates in the time-varying communication context. The detailed updates are summarized in Algorithm 1. Before proceeding, we introduce the definitions and assumptions made in this paper.

1) Definitions and Assumptions:

Definition 1. *B-Connectivity: A time-varying graph is said to be B-connected when the union of its edge sets are connected over a time period of length B. Mathematically this can be written as:*

$$(\mathcal{V}, \mathcal{E}_t \cup \mathcal{E}_{t+1} \cup \cdots \cup \mathcal{E}_{t+(B-1)}),$$

where $t, B \in \mathbb{N}$. Intuitively, this can be seen as a sliding window of length B for which the edge sets contained within this window are connected. This is visualized in Figure 1.

Definition 2. *The neighborhood of agent i for a time t is defined as $\mathcal{N}^i_t = \{j | (i,j) \in \mathcal{E}_t, i, j \in \mathcal{V}\}$.*

Definition 3. *We define the mixing time as $\tau(\beta) \geq 1$ that satisfies*

$$\left\|\bar{A} - \mathbb{E}[A(X_k)|X_0 = i]\right\| \leq \beta, \quad \forall i, \forall k \geq \tau(\beta) \tag{3}$$

$$\|\mathbb{E}[b(X_k)|X_0 = i]\| \leq \beta, \quad \forall i, \forall k \geq \tau(\beta) \tag{4}$$

where $\bar{A}, A(X_k)$, and $b(X_k)$ are defined as follows:

$$\bar{A} = \begin{bmatrix} -1 & 0 \\ -\Phi^\top D^s \mathbf{1} & \Phi^\top D^s (P^\pi - I)\Phi \end{bmatrix}$$

$$A(X_k) = \begin{bmatrix} -1 & 0 \\ -\phi(Z_t) & \phi(Z_t)(\phi(Z_{t+1}) - \phi(Z_t))^\top \end{bmatrix}$$

$$b(X_k) = \begin{bmatrix} r_{t+1} \\ \phi(Z_t) r_{t+1} \end{bmatrix} - A(X_k) \begin{bmatrix} J_\pi \\ w^\star \end{bmatrix}.$$

The mixing time $\tau(\beta)$ is the minimum time required for Inequalities (3) *and* (4) *to hold true. More details of this can be found in [17].*

Assumption 1. *The graph $\mathcal{G}_t$, as defined in Section III, is B-connected as given by Definition 1.*

Assumption 2. *It is assumed that the induced Markov chain $\{s_t\}_{t>0}$ for a given policy π is aperiodic and irreducible.*

Assumption 3. *The rewards for each agent $i \in \mathcal{V}$ are bounded by $|r^i| \leq r_{\max},\ \forall t \geq 0$.*

Assumption 4. *The adjacency matrices $A(t) \in \mathbb{R}^{N\times N}$ at each time t representing the graphs $\mathcal{G}_t$ are doubly stochastic—the elements of each row are non-negative, and they sum to one for the matrices A and $A^\top$. The edges must have weights that satisfy $[A(t)]_i^i{\geq}\eta, \forall i \in \mathcal{V}$ where $\eta > 0$. If there is communication between agents i and j then $[A(t)]_j^i{\geq}\eta$, otherwise $[A(t)]_j^i{=}0$.*

Assumption 5. *The value function can be approximated by a linear function, parameterized by the weight vector w; specifically,*

$$V(s;w) \approx \phi(s)^\top w,$$

where the feature vector $\phi(s) = [\phi_1(s)\cdots\phi_n(s)]^\top{\in}\mathbb{R}^n$ and $\|\phi(s)\| \leq 1$. The feature vector dimension is usually $n < |\mathcal{S}|$ since the goal is to approximate a value function for a large state space. From these feature vectors, one can construct a matrix $\Phi \in \mathbb{R}^{|\mathcal{S}|\times n}$, where $\nexists u \in \mathbb{R}^n$ such that $\Phi u \neq \mathbf{1}$, to ensure that all the states do not have the same value. In the case where $\Phi u \neq \mathbf{1}$, this means that all the states are approximated to have the same value; therefore, no particular state is more beneficial than another and nothing is gained from this using this value function.

Assumption 1 upper bounds the time interval during which the communication network is possibly disconnected. This assumption enables the convergence analyses since the setting is fully distributed, requiring inter-agent interaction to solve the policy evaluation problem. Assumption 2 is needed to establish the existence of a unique stationary distribution, Assumptions 3 and 5 are standard in reinforcement learning literature [2], [18]; and Assumption 4 is needed to ensure balanced distribution of computation and updates, and for symmetry in information sharing. Finally, we make Assumption 5 since linear function approximation simplifies theoretical analyses, and provides insight on the behavior of the algorithm.

Algorithm 1 TD Learning with Local and Intermittent Consensus Updates

```
Input: Initial state $s_0$, feature map $\phi(s)$, initial parameters $\{w^i_{L,0}\}$, step size $\beta$, communication round $L$, local step number $K$
for $l = 0$ to $L-1$ do
    $s_{l,0} \leftarrow s_{l-1,K}$
    for $k = 0$ to $K-1$ do
        for all agents $i \in N$ do
            Execute action $a_i \sim \pi^i(\cdot|s_{l,k})$
            Observe state $s_{l,k+1}$ and reward $r^i_{l,k+1}$
            $\delta^i_{l,k} \leftarrow r^i_{l,k+1} + \phi(s_{l,k+1})^T w_i - \mu^i_{l,k} - \phi(s_{l,k})^T w_i$
            $\mu^i_{l,k+1} \leftarrow (1-\beta)\mu^i_{l,k} + \beta r^i_{l,k+1}$
            Locally Update Weight Vectors:
            $w^i_{l,k+1} \leftarrow w^i_{l,k} + \beta\delta^i_{l,k}\cdot\phi(s_{l,k})$
        end for
    end for
    for all agents $i \in N$ do
        Consensus update: $w^i_{l+1,0} \leftarrow \sum_{j\in N_i(t)} A(t)_{ij} w^j_{l,K}$
    end for
end for
Output: Parameters $w^i_{L,0} \in \mathcal{V}$
```

IV. Main Result and Convergence Analysis

Based on Algorithm 1, our main result is summarized in Theorem 1, which we prove via a series of lemmas.

Theorem 1. *Given Assumptions 1-5 and Algorithm 1. Let $\rho \equiv (1+4\beta K)(1-\eta^{(N-1)B})^{\frac{1}{(N-1)B}}$. If*

$$0 < \beta K < \min\left\{\frac{1}{2}, \frac{1-(1-\eta^{B_0})^{\frac{1}{B_0}}}{4(1-\eta^{B_0})^{\frac{1}{B_0}}}\right\},$$

then $0 < \rho < 1$ and the difference between the aggregate deviation of the agents' weight vectors and the optimal weight vector $w^\star$ is upper bounded by

$$\mathbb{E}\Big[\sum_{i=1}^{N} \big\|w^i_{L,0} - w^\star\big\|^2\Big] \leq 2\Big(\frac{\kappa_1 \rho^L}{(1-\eta^{B_0})^{\frac{1}{B_0}}}\|Q_{0,0}\| + \frac{\kappa_2 \beta K}{1-\rho}\Big)^2$$
$$+ 2N\Big(c_3\beta\tau(\beta) + c_2(1-c_1\beta)^{KL-\tau(\beta)}\Big(\sqrt{\|\bar{w}_0 - w^\star\|^2 + (\mu_0 + J_\pi)^2} + \frac{r_{\max}}{3}\Big)^2\Big), \tag{5}$$

where $Q_{L,0} \in \mathbb{R}^{n\times N}$ is a matrix that contains N columns $w^i_{L,0} - \bar{w}_{L,0}$ and $B_0 = (N-1)B$ for which B is the same as mentioned in Assumption 1. The consensus matrix for each time-step k and communication round l is defined as shown below.

$$Q_{l,k} \equiv \begin{bmatrix} | & & | \\ (w^1_{l,k} - \bar{w}_{l,k}) & \cdots & (w^N_{l,k} - \bar{w}_{l,k}) \\ | & & | \end{bmatrix}$$

Remark 1. *Theorem 1 shows that each of the individual agents' weight vectors reaches within some region of the optimal value. Interestingly, the two terms of Equation* (5) *show the dependence of the convergence of Algorithm 1 on the consensus of the agents and the dynamics of the average behavior in relation to the optimal solution $w^\star$. Furthermore, the first term in the second inequality shows, as one would intuitively expect, that the consensus error decreases with more inter-agent communication rounds L when the condition on βK is satisfied.*

A. Proof of Theorem 1 (Proof Sketch)

Proof. Theorem 1 is proved by showing that the agents' individual weights $w^i_{l,k}$ converge to a consensus weight vector $\bar{w}_{L,0}$ as given by [1], which converges to the optimal weight $w^\star$; that is,

$$\mathbb{E}\Big[\sum_{i=1}^{N} \big\|w^i_{L,0} - w^\star\big\|^2\Big] = \mathbb{E}\Big[\sum_{i=1}^{N} \big\|w^i_{L,0} - \bar{w}_{L,0} + \bar{w}_{L,0} - w^\star\big\|^2\Big] \leq 2\mathbb{E}\Big[\sum_{i=1}^{N} \big\|w^i_{L,0} - \bar{w}_{L,0}\big\|^2\Big] + 2\mathbb{E}\Big[\sum_{i=1}^{N} \|\bar{w}_{L,0} - w^\star\|^2\Big]$$
$$= 2\mathbb{E}\Big[\|Q_{L,0}\|_F^2\Big] + 2N\mathbb{E}\Big[\|\bar{w}_{L,0} - w^\star\|^2\Big]. \tag{6}$$

Our analysis focuses on the convergence of the agents' individual weights to an average weight, since that is the part directly affected by the time-varying communication assumption. To accomplish this, we obtain a bound on the consensus errors next. Additionally, it should be noted that Equations (5) and (6) are equivalent, but the latter is simply a compact representation of the former to gain intuition on what each of the terms represent. ■

B. Consensus error

Lemma 1. *The consensus matrix defined in Theorem 1 is given by*

$$Q_{L,0} = \prod_{n=0}^{L-1}\prod_{m=0}^{K-1} B_{n,m} Q_{0,0} \prod_{p=1}^{L} A^\top(p) + \sum_{q=0}^{L-1}\prod_{j=1}^{L-1-q}\prod_{k=0}^{K-1} B_{q+j,k} \sum_{t=0}^{K-1}\prod_{\tilde{t}>t}^{K-1} B_{q,\tilde{t}} C_{q,t}(\mathrm{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^\top)\prod_{l=1}^{L-q} A^\top(l), \tag{7}$$

where $B_{n,m} = I + \beta\phi(s_{n,m})[\phi(s_{n,m+1}) - \phi(s_{n,m})]^\top$, $c^i_{l,k} = \beta(r^i_{l,k+1} - \mu^i_{l,k})\phi(s_{l,k})$, and $C_{n,m} = [c^1_{n,m} \ldots c^N_{n,m}]$ are respectively functions of the feature vectors $\phi(\cdot)$, exponentially weighted average reward μ^i, and individual rewards r^i.

Lemma 2. *The magnitude of the consensus error matrix is upper bounded by FIRST check out the limits on network sizes for time-varying networks. How large would the network have to be to prevent convergence?*

$$\|Q_{L,0}\| \leq \frac{\kappa_1 \rho^L}{(1-\eta^{B_0})^{\frac{1}{B_0}}}\|Q_{0,0}\| + \frac{\kappa_2 \beta K}{1-\rho}, \tag{8}$$

where $\kappa_1 = 2N\frac{1+\eta^{-B_0}}{1-\eta^{B_0}}$ and $\kappa_2 = 8N^{3/2}r_{\max}(1+\eta^{-B_0})/(1-\eta^{B_0})$. Given that the bounds for βK are met as stated in Theorem 1, $0 < \rho < 1$, which means that as the number of consensus updates increases, the tighter the error bounds will become check.

Remark 2. *A bound for the first summand in the RHS of Equation* (8) *is obtained in Lemma 3 in Appendix A. A bound on the second summand has been obtained in [1].*

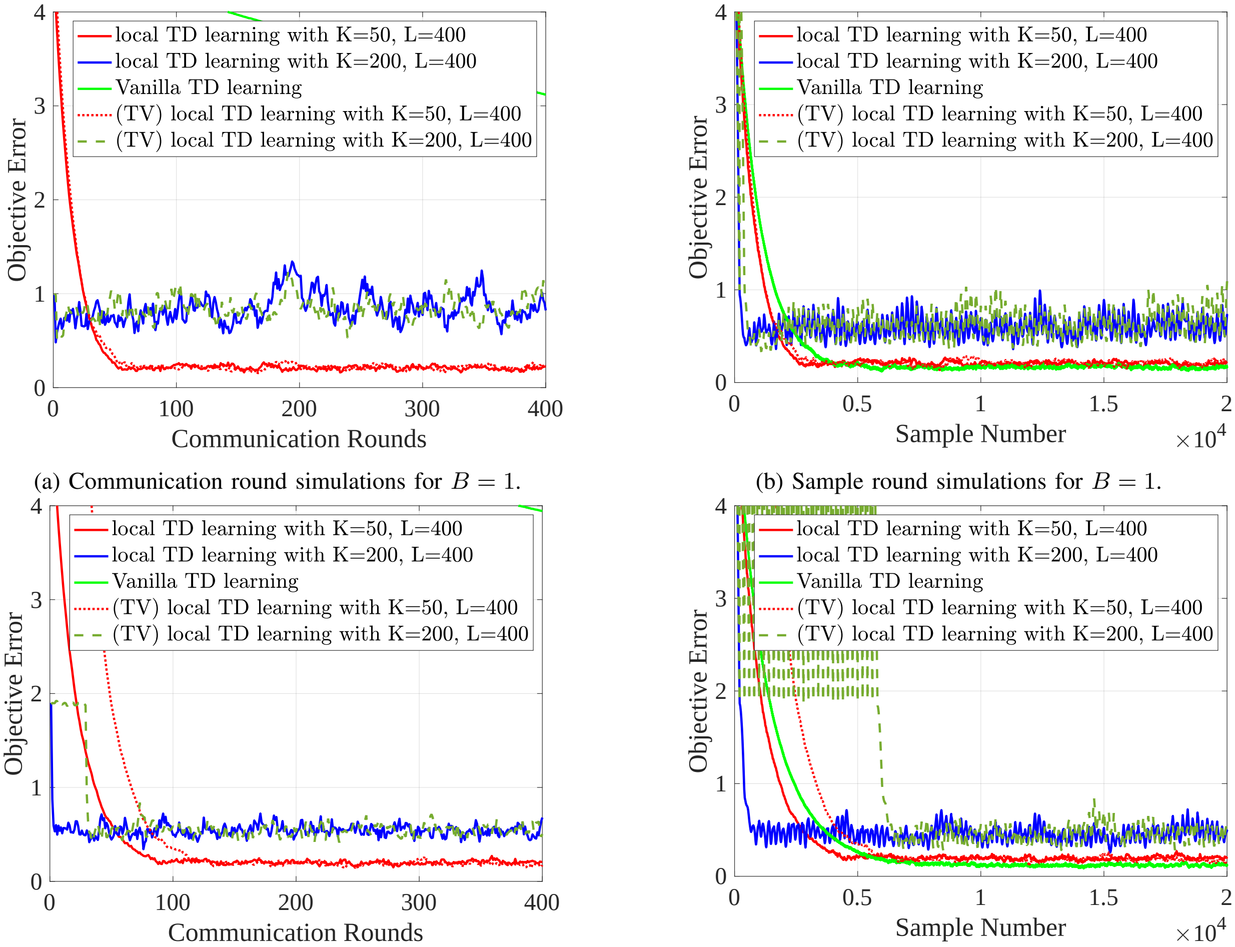
Theorem 2. *The upper bound on the expected value of the difference between the optimal weight vector and the average of the weight vectors is given by*

$$\mathbb{E}[\|\bar{w}_{L,0} - w^\star\|^2] \le c_2(1-c_1\beta)^{KL-\tau(\beta)}\left(\sqrt{\|\bar{w}_0 - w^\star\|^2 + (\mu_0+J_\pi)^2}+\frac{r_{\max}}{3}\right)^2 + c_3\beta\tau(\beta), \tag{9}$$

where $c_1, c_2,$ and c_3 are defined in [1], [17].

The proof of Theorem 2 is omitted since it follows a similar proof in [1].

V. Experimentation and Validation



(a) Communication round simulations for $B = 1$.

(b) Sample round simulations for $B = 1$.

(c) Communication round simulations for $B = 15$.

(d) Sample round simulations for $B = 15$.

Fig. 2: Simulations showing the effects of communication and sampling for different B values to satisfy Assumption 1. The TV lines represent the case in which the communication network is time-varying (TV).

To experimentally validate the theoretical results, simulations were carried out using synthetic data, with a framework very similar to the work done in [1] to be able to compare the performance and outcomes. A simple environment was initialized with $N = 15$ agents, each with binary action spaces $\mathcal{A}^i = \{0, 1\}$, and an entire state-space of cardinality $|\mathcal{S}| = 10$. Additionally, the dimension of the feature vector chosen was $d = 5$ and The transition matrix was generated by first randomly generating an $|\mathcal{S}| \times |\mathcal{S}|$ matrix with elements being sampled from a uniform distribution with a domain of $[0, 1]$. After the creation of the random matrix, all the rows were normalized be stochastic. To generate "real-time" rewards for agents, first, random numbers were generated from a uniform distribution on the interval $[0, 4]$ for each $r^i(s, a)$. Next, while the simulations were run, the instantaneous reward for an agent were generated from the uniform distribution $[r^i(s, a) - 0.5, r^i(s, a) + 0.5]$. Since the policy being used for PE is not required to be optimal, a simple policy of $\pi^i(\cdot|s) = 0.5$ was set for all agents and states. Similarly, the feature matrix mentioned in Assumption 5 was created by randomly generating $|\mathcal{S}| \times n$ values that make linearly independent feature vectors, that allow for a Φ such that $\Phi u \neq \mathbf{1}$, and are of unit length. Finally, the objective error was calculated as

$$\text{OE=averaged over 10 rounds } \frac{\sqrt{\sum_{i=1}^{N} \left\| w_{l,k}^{i} - w^{\star} \right\|^{2}}}{nN},$$

where

$$w^{\star} = -\Psi^{-1} b, \tag{10}$$

as given by Equations (5) and (6) in [1].

In contrast to [1] and in line with Assumption 1, the adjacency matrices were created in such a way that the union of every B consecutive graph is connected. Due to the various constraints set on the network matrices, some of them had to be relaxed such as the double stochasticity, symmetry, and the lower bounding of the matrix elements—this was done by allowing the matrices to be generated so that $A - A^{\top} \approx 0$ where A is a randomly generated matrix that is said to be approximately symmetric.

From the figures above, the most important takeaway is that the local TD algorithm presented in [1] has similar behavior for both the static and time-varying (TV) communication network. What differentiates both settings is when $B > 1$ causes a delay in the learning process as shown in Figure 2d. Besides that, we can see that the error is higher on average when K is increased as expected because there is more time for the weight parameters to drift away from the true value.

VI. Concluding Remarks

This paper studied the problem of policy evaluation for a team of fully distributed learners using a distributed temporal difference learning over time-varying communication networks. The algorithm presented and analyzed effectively addresses the challenges posed by a dynamic communication topology in agents cooperatively evaluating their policies to improve estimation of their value functions. The approach presented is scalable, allowing for collaboration among multiple agents while maintaining responsiveness to changes in the network topology. The integration of temporal difference learning enhances the accuracy of value estimates, facilitating improved decision-making processes. Theoretical results are validated to show that the consensus-based scheme is sample efficient.

Appendix

A. Proof of Lemma 1

Proof. We will use an inductive proof. Starting with our basis step we get

$$\begin{aligned} Q_{1,0} &= \prod_{m=0}^{K-1} B_{0,m} Q_{0,0} A^{\top}(1) + \sum_{t=0}^{K-1} \prod_{\tilde{t}>t}^{K-1} B_{0,\tilde{t}} C_{0,t} (\mathrm{I} - \frac{1}{N}\mathbf{11}^{\top}) A^{\top}(1) \\ &= \left[\prod_{m=0}^{K-1} B_{0,m} Q_{0,0} + \sum_{t=0}^{K-1} \prod_{\tilde{t}>t}^{K-1} B_{0,\tilde{t}} C_{0,t} (\mathrm{I} - \frac{1}{N}\mathbf{11}^{\top}) \right] A^{\top}(1) = Q_{0,K} A^{\top}(1). \end{aligned}$$

This shows that the base case for our equation is true. Next, we will assume that the following equality is true

$$Q_{L,0} = \prod_{n=0}^{L-1} \prod_{m=0}^{K-1} B_{n,m} Q_{0,0} \prod_{p=1}^{L} A^{\top}(p) + \sum_{q=0}^{L-1} \prod_{j=1}^{L-1-q} \prod_{k=0}^{K-1} B_{q+j,k} \sum_{t=0}^{K-1} \prod_{\tilde{t}>t}^{K-1} B_{q,\tilde{t}} C_{q,t} (\mathrm{I} - \frac{1}{N}\mathbf{11}^{\top}) \prod_{m=1}^{L-q} A^{\top}(m). \tag{11}$$

The inductive step is then performed by looking at the case of $Q_{L+1,0}$.

$$Q_{L+1,0} = \prod_{n=0}^{L} \prod_{m=0}^{K-1} B_{n,m} Q_{0,0} \prod_{p=1}^{L+1} A^{\top}(p) + \sum_{q=0}^{L} \prod_{j=1}^{L-q} \prod_{k=0}^{K-1} B_{q+j,k} \sum_{t=0}^{K-1} \prod_{\tilde{t}>t}^{K-1} B_{q,\tilde{t}} C_{q,t} (\mathrm{I} - \frac{1}{N}\mathbf{11}^{\top}) \prod_{m=1}^{L+1-q} A^{\top}(m)$$

Pulling out one of the summed terms from the second term.

$$\begin{aligned} Q_{L+1,0} = &\prod_{n=0}^{L} \prod_{m=0}^{K-1} B_{n,m} Q_{0,0} \prod_{p=1}^{L+1} A^{\top}(p) + \sum_{q=0}^{L-1} \prod_{j=1}^{L-q} \prod_{k=0}^{K-1} B_{q+j,k} \sum_{t=0}^{K-1} \prod_{\tilde{t}>t}^{K-1} B_{q,\tilde{t}} C_{q,t} (\mathrm{I} - \frac{1}{N}\mathbf{11}^{\top}) \prod_{m=1}^{L+1-q} A^{\top}(m) \\ &+ \sum_{t=0}^{K-1} \prod_{\tilde{t}>t}^{K-1} B_{L,\tilde{t}} C_{L,t} (\mathrm{I} - \frac{1}{N}\mathbf{11}^{\top}) A^{\top}(L+1) \end{aligned}$$

Factoring out $A^\top(L+1)$ and $B_{L,m}$ product.

$$\begin{aligned} Q_{L+1,0}=&\Bigg[\prod_{m=0}^{K-1} B_{L,m}\Bigg[\prod_{n=0}^{L-1}\prod_{m=0}^{K-1} B_{n,m}Q_{0,0}\prod_{p=1}^{L}A^\top(p)+\sum_{q=0}^{L-1}\prod_{j=1}^{L-1-q}\prod_{k=0}^{K-1}B_{q+j,k}\sum_{t=0}^{K-1}\prod_{\tilde{t}>t}^{K-1}B_{q,\tilde{t}}C_{q,t}(\mathrm{I}-\frac{1}{N}\mathbf{11}^\top)\prod_{m=1}^{L-q}A^\top(m)\Bigg] \\ &+\sum_{t=0}^{K-1}\prod_{\tilde{t}>t}^{K-1}B_{L,\tilde{t}}C_{L,t}(\mathrm{I}-\frac{1}{N}\mathbf{11}^\top)\Bigg]A^\top(L+1)\end{aligned}$$

Remembering the assumed equality given by Equation (11), we are able to write

$$Q_{L+1,0}=\left[\prod_{m=0}^{K-1} B_{L,m}Q_{L,0}+\sum_{t=0}^{K-1}\prod_{\tilde{t}>t}^{K-1}B_{L,\tilde{t}}C_{L,t}(\mathrm{I}-\frac{1}{N}\mathbf{11}^\top)\right]A^\top(L+1).$$

Finally, this can be written as

$$Q_{L+1,0}=Q_{L,K}A^\top(L+1)$$

which agrees with the standard consensus update equation. ■

B. Proof of Lemma 3

Before moving to the next result, we show that by taking the norm of both the left and right-hand sides of eq. (7), we obtain

$$\|Q_{L,0}\|_F=\left\|\prod_{n=0}^{L-1}\prod_{m=0}^{K-1} B_{n,m}Q_{0,0}\prod_{p=1}^{L}A^\top(p)\ +\sum_{q=0}^{L-1}\prod_{j=1}^{L-1-q}\prod_{k=0}^{K-1}B_{q+j,k}\sum_{t=0}^{K-1}\prod_{\tilde{t}>t}^{K-1}B_{q,\tilde{t}}C_{q,t}(\mathrm{I}-\frac{1}{N}\mathbf{11}^\top)\prod_{m=1}^{L-q}A^\top(m)\right\|_F.$$

Now using the triangle inequality

$$\|Q_{L,0}\|_F\leq\left\|\prod_{n=0}^{L-1}\prod_{m=0}^{K-1} B_{n,m}Q_{0,0}\prod_{p=1}^{L}A^\top(p)\right\|+\left\|\sum_{q=0}^{L-1}\prod_{j=1}^{L-1-q}\prod_{k=0}^{K-1}B_{q+j,k}\sum_{t=0}^{K-1}\prod_{\tilde{t}>t}^{K-1}B_{q,\tilde{t}}C_{q,t}(\mathrm{I}-\frac{1}{N}\mathbf{11}^\top)\prod_{m=1}^{L-q}A^\top(m)\right\|_F. \tag{12}$$

Lemma 3. *The following inequality holds where the LHS is the first terms of Equation* (12)*:*

$$\left\|\prod_{n=0}^{L-1}\prod_{m=0}^{K-1} B_{n,m}Q_{0,0}\prod_{p=1}^{L}A^\top(p)\right\|_F\leq\frac{\kappa_1\rho^L}{(1-\eta^{B_0})^{\frac{1}{B_0}}}\|Q_{0,0}\|_F.$$

Proof. Using the Cauchy-Schwarz inequality,

$$\left\|\prod_{n=0}^{L-1}\prod_{m=0}^{K-1} B_{n,m}Q_{0,0}\prod_{p=1}^{L}A^\top(p)\right\|_F\leq\left\|\prod_{n=0}^{L-1}\prod_{m=0}^{K-1} B_{n,m}\right\|_F\cdot\left\|Q_{0,0}\prod_{p=1}^{L}A^\top(p)\right\|_F \tag{13}$$

For simplicity of notation, we will define the transition matrix to be $\Phi(L,1)$. Specifically, $\prod_{p=1}^{L}A^\top(p)=A^\top(1)\cdots A^\top(L)=\Phi(L,1)$. More generally, $\Phi(k,s)=A^\top(s)A^\top(s+1)\cdots A^\top(k-1)A^\top(k)$ where $k>s$ and $k,s\in\mathbb{N}$. Looking at the second norm on the right-hand side of the inequality previously shown, we get $\left\|Q_{0,0}\prod_{p=1}^{L}A^\top(p)\right\|_F=\|Q_{0,0}\Phi(L,1)\|_F$. For clarification, it should be noted that the brackets with the superscript represent the jth column of a matrix

$$\|Q_{0,0}\Phi(L,1)\|_F^2=\sum_{j=1}^{N}\left\|Q_{0,0}[\Phi(L,1)]^j\right\|_2^2. \tag{14}$$

Focusing on the norm in Equation (14) and using the result from [19] that $\bar{\Phi}(s)=\lim_{k\to\infty}\Phi(k,s)=\frac{1}{N}\mathbf{11}^\top$

$$\begin{aligned}\left\|Q_{0,0}[\Phi(L,1)]^j\right\|_2^2&=\left\|Q_{0,0}([\Phi(L,1)]^j-[\bar{\Phi}(1)]^j)\right\|_2^2\\ &=\sum_{i=1}^{n}\left|[Q_{0,0}]_i\Big([\Phi(L,1)]^j-\frac{1}{N}\mathbf{1}\Big)\right|^2.\\ &\leq\sum_{i=1}^{n}\|[Q_{0,0}]_i\|_2^2\left\|[\Phi(L,1)]^j-\frac{1}{N}\mathbf{1}\right\|_2^2\end{aligned}$$

After using the Cauchy-Schwarz inequality once again, the results from [19] can be used to obtain

$$\begin{aligned}\left\|[\Phi(L,1)]^j - \frac{1}{N}\mathbf{1}\right\|_2^2 = \sum_{l=1}^{N}\left|[\Phi(L,1)]_l^j - \frac{1}{N}\right|^2 &\leq \sum_{l=1}^{N}\Big(2\frac{1+\eta^{-B_0}}{1-\eta^{B_0}}(1-\eta^{B_0})^{(L-1)/B_0}\Big)^2 \\ &= 4N\Big(\frac{1+\eta^{-B_0}}{1-\eta^{B_0}}\Big)^2(1-\eta^{B_0})^{2(L-1)/B_0}.\end{aligned}$$

Using this result we get the inequality below after some algebra

$$\left\|Q_{0,0}([\Phi(L,1)]^j)\right\|_2 \leq 2\sqrt{N}\Big(\frac{1+\eta^{-B_0}}{1-\eta^{B_0}}\Big)(1-\eta^{B_0})^{(L-1)/B_0}\|Q_{0,0}\|_F.$$

Finally, we can write

$$\|Q_{0,0}\Phi(L,1)\|_F \leq 2N\Big(\frac{1+\eta^{-B_0}}{1-\eta^{B_0}}\Big)(1-\eta^{B_0})^{(L-1)/B_0}\|Q_{0,0}\|_F$$

Similar to [1], we will let $\kappa_1 = 2N\frac{1+\eta^{-B_0}}{1-\eta^{B_0}}$ which means that

$$\|Q_{0,0}\Phi(L,1)\|_F \leq \kappa_1(1-\eta^{B_0})^{(L-1)/B_0}\|Q_{0,0}\|_F. \tag{15}$$

Since Equation (15) gives us the bounds for the second norm taken in Equation (13) and the bounds for the first norm are given in [1], we can write the inequality as

$$\begin{aligned}\left\|\prod_{n=0}^{L-1}\prod_{m=0}^{K-1}B_{n,m}Q_{0,0}\prod_{p=1}^{L}A^\top(p)\right\|_F &\leq \kappa_1(1+2\beta)^{KL}(1-\eta^{B_0})^{(L-1)/B_0}\|Q_{0,0}\|_F \\ &\leq \kappa_1(1+4\beta K)^L(1-\eta^{B_0})^{(L-1)/B_0}\|Q_{0,0}\|_F \\ &= \kappa_1\rho^L(1-\eta^{(N-1)B})^{\frac{-1}{(N-1)B}}\|Q_{0,0}\|_F = \frac{\kappa_1\rho^L}{(1-\eta^{B_0})^{\frac{1}{B_0}}}\|Q_{0,0}\|_F.\end{aligned}$$

Where the inequality below, which is valid for small x, was used to get the second inequality above.

$$(1+x)^K \leq 1+2xK$$

This also means that for $\rho \equiv (1+4\beta K)(1-\eta^{(N-1)B})^{\frac{1}{(N-1)B}}$ to be $0<\rho<1$, the following must be true:

$$0 < \beta K < \min\left\{\frac{1}{2}, \frac{1-(1-\eta^{B_0})^{\frac{1}{B_0}}}{4(1-\eta^{B_0})^{\frac{1}{B_0}}}\right\}.$$

The $1/2$ is included in the bounds for βK because it is required for a Taylor series expansion that was used in the bounding for the norm with $B_{n,m}$. ■

References

[1] F. Hairi, Z. Zhang, and J. Liu, "Sample and communication efficient fully decentralized marl policy evaluation via a new approach: Local td update," *arXiv preprint arXiv:2403.15935*, 2024.

[2] T. Doan, S. Maguluri, and J. Romberg, "Finite-time analysis of distributed td (0) with linear function approximation on multi-agent reinforcement learning," in *International Conference on Machine Learning*, PMLR, 2019, pp. 1626–1635.

[3] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Basar, "Fully decentralized multi-agent reinforcement learning with networked agents," in *International conference on machine learning*, PMLR, 2018, pp. 5872–5881.

[4] J. Orr and A. Dutta, "Multi-agent deep reinforcement learning for multi-robot applications: A survey," *Sensors*, vol. 23, no. 7, p. 3625, 2023.

[5] C. Hu, G. Wen, S. Wang, J. Fu, and W. Yu, "Distributed multiagent reinforcement learning with action networks for dynamic economic dispatch," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[6] H.-T. Wai, Z. Yang, Z. Wang, and M. Hong, "Multi-agent reinforcement learning via double averaging primal-dual optimization," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[7] Z. Zhang and D. Wang, "Adaptive individual q-learning—a multiagent reinforcement learning method for coordination optimization," *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[8] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," *Advances in neural information processing systems*, vol. 12, 1999.

[9] Z. Zhang, Y.-S. Ong, D. Wang, and B. Xue, "A collaborative multiagent reinforcement learning method based on policy gradient potential," *IEEE transactions on cybernetics*, vol. 51, no. 2, pp. 1015–1027, 2019.

[10] D. K. Kim, M. Liu, M. D. Riemer, *et al.*, "A policy gradient algorithm for learning to learn in multiagent reinforcement learning," in *International Conference on Machine Learning*, PMLR, 2021, pp. 5541–5550.

[11] R. S. Sutton and A. G. Barto, "Temporal-difference learning," *Reinforcement learning: an introduction*, pp. 167–200, 1998.

[12] A. Nedic, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM Journal on Optimization*, vol. 27, no. 4, pp. 2597–2633, 2017.

[13] H. Li, Q. Lü, X. Liao, and T. Huang, "Accelerated convergence algorithm for distributed constrained optimization under time-varying general directed graphs," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 7, pp. 2612–2622, 2018.

[14] A. Simonetto, E. Dall'Anese, S. Paternain, G. Leus, and G. B. Giannakis, "Time-varying convex optimization: Time-structured algorithms and applications," *Proceedings of the IEEE*, vol. 108, no. 11, pp. 2032–2048, 2020.

[15] J. Zhu, B. Li, L. Wang, *et al.*, "Provable distributed adaptive temporal-difference learning over time-varying networks," *Expert Systems with Applications*, vol. 228, p. 120 406, 2023.

[16] J. Tsitsiklis and B. Van Roy, "Analysis of temporal-diffference learning with function approximation," *Advances in neural information processing systems*, vol. 9, 1996.

[17] R. Srikant and L. Ying, "Finite-time error bounds for linear stochastic approximation andtd learning," in *Conference on Learning Theory*, PMLR, 2019, pp. 2803–2830.

[18] L.-y. Zhao, T.-q. Chang, L. Zhang, X.-l. Zhang, and J.-f. Wang, "Multi-agent cooperation policy gradient method based on enhanced exploration for cooperative tasks," *International Journal of Machine Learning and Cybernetics*, vol. 15, no. 4, pp. 1431–1452, 2024.

[19] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.