IEEE Control Systems Letters paper presented at
2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Distributed TD(0) with Almost No Communication

Rui Liu[1] and Alex Olshevsky[2]

Abstract— **We provide a new non-asymptotic analysis of distributed temporal difference learning with linear function approximation. Our approach relies on "one-shot averaging," where N agents run identical local copies of the TD(0) method and average the outcomes only once at the very end. We demonstrate a version of the linear time speedup phenomenon, where the convergence time of the distributed process is a factor of N faster than the convergence time of TD(0). This is the first result proving benefits from parallelism for temporal difference methods.**

I. Introduction

Recent years have seen reinforcement learning used in a variety of multi-agent systems. However, a rigorous understanding of how standard methods in reinforcement learning perform in a multi-agent setting with limited communication is only beginning to be available.

One of the most fundamental problems in reinforcement learning is policy evaluation, and one of the most basic policy evaluation algorithms is temporal difference (TD) learning, originally proposed in [18]. TD learning works by updating a value function from differences in predictions over a succession of steps in the underlying Markov Decision Process (MDP).

Developments in the field of multi-agent reinforcement learning have led to an increased interest in decentralizing TD methods, which is the subject of this paper. We will consider a simple model where N agents all have access to their own copy of the same MDP. Naturally, the agents can simply ignore each other and run any policy evaluation method without communication. However, this ignores the possibility that agents can benefit from mixing local computations by each agent and inter-agent interactions. Our goal will be to quantify how much TD methods can benefit from this.

A. Related Literature

A natural benchmark to compare the performance of distributed TD methods to is the performance of centralized TD methods. Precise conditions for the asymptotic convergence result were first given in [20] by viewing TD as a stochastic approximation for solving a Bellman equation. Recently, there has been an increased interest in non-asymptotic convergence results, e.g., [3], [2]. The state of the art results show that, under i.i.d samples, TD algorithm with linear function approximation converges with rates of $O(1/\sqrt{T})$ for value function with step-size $1/\sqrt{T}$ and converge as fast as $O(1/t)$ with step-size $O(1/t)$ [2].

Prior to this work, there have been several analyses of distributed TD with linear function approximation [17], [5], [21]. However, the model considered by those papers is very different than the model considered here, as those papers considered agents interacting collectively with an environment with a transition function that depends on all the actions taken by the agents. This is a much more difficult setting than what we consider in this paper, where we have N MDPs which are completely decoupled, except insofar as the agents may choose to couple them via an exchange of messages.

Perhaps the most relevant previous work is [16] which addresses actor-critic rather than temporal difference methods. It is shown there, up to a certain approximation error, it is possible to obtain a speedup proportional to the number of nodes for a distributed model of actor-critic. Besides [16], another example of a similar result we are aware of is [8]. However, [8] came after the present work (note that [8] cites the arxiv version [10] of the present paper, which appeared on the arxiv about a year before the arxiv version of [8]). The paper [8] considers the much more general problem of distributed (or federated) stochastic approximation, which includes temporal difference learning as studied here, alongside many other problems (such as Q-learning). A linear speedup is obtained similar to our results here, but it requires N averages throughout the course of the algorithm – in contrast to the single averaging round required in this work.

B. Our contributions

We show a version of a "linear speedup" phenomenon: under a number of assumptions, we show that the convergence bounds of a distributed algorithm with N agents is a factor of N faster than the corresponding convergence time bounds associated with a centralized version. To our knowledge, this is the first example of this phenomenon being demonstrated in reinforcement learning.

These results arguably justify the introduction of our model in this paper, which should be contrasted with the much harsher models considered in the previous multi-agent reinforcement learning literature. Indeed, the model presented here allows for the possibility of speeding up reinforcement learning by parallelizing computations.

II. Preliminaries

We begin by standardizing notation and providing standard background information on Markov Decision Processes and temporal difference methods.

A. Markov Decision Processes

A discounted reward MDP is described by a 5-tuple $(\mathcal{S},\mathcal{A},\mathcal{P},r,\gamma)$, where $\mathcal{S}=[n]=\{1,2,\cdots,n\}$ is a finite state

[1] Rui Liu is with the Division of Systems Engineering, Boston University, Boston, MA, USA rliu@bu.edu

[2] Alex Olshevsky is with the Department of ECE and Division of Systems Engineering, Boston University, Boston, MA, USA alexols@u.edu

space, $\mathcal{A}$ is a finite action space, $\mathcal{P}(s'|s,a) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0,1]$ is transition probability from s to s' determined by a, $r(s,a,s') : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ are deterministic rewards and $\gamma \in (0,1)$ is the discount factor.

Let μ denote a fixed policy that maps a state $s \in \mathcal{S}$ to a probability distribution $\mu(\cdot|s)$ over the action space $\mathcal{A}$, so that $\sum_{a \in \mathcal{A}} \mu(a|s) = 1$. For such a fixed policy μ, define the instantaneous reward vector $R^{\mu} : \mathcal{S} \to \mathbb{R}$ as

$$R^{\mu}(s) = \sum_{s' \in \mathcal{S}} \sum_{a \in \mathcal{A}} \mu(a|s) \mathcal{P}(s'|s,a) r(s,a,s').$$

Fixing the policy μ induces a probability transition matrix between states:

$$P^{\mu}(s,s') = \sum_{a \in \mathcal{A}} \mu(a|s) \mathcal{P}(s'|s,a).$$

We will use $r_t = r(s_t, a_t, s_{t+1})$ to denote the instantaneous reward at time t, where s_t, a_t are the state and action taken at step t. The value function of μ, denoted by $V^{\mu} : \mathcal{S} \to \mathbb{R}$ is defined as $V^{\mu}(s) = E_{\mu,s}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$, where $E_{\mu,s}[\cdot]$ indicates that s is the initial state and the actions are chosen according to the policy μ. In the following, we will treat V^{μ} and R^{μ} as vectors in $\mathbb{R}^n$ and treat P^{μ} as a matrix in $\mathbb{R}^{n \times n}$.

Next, we state a standard assumption on the underlying Markov chain.

Assumption 1. *The Markov chain with transition matrix P^{μ} is irreducible and aperiodic.*

A consequence of Assumption 1 is that there exists a unique stationary distribution $\pi = (\pi_1, \pi_2, \cdots, \pi_n)$, a row vector whose entries are positive and sum to 1. This stationary distribution satisfies $\pi^T P^{\mu} = \pi^T$ and $\pi_{s'} = \lim_{t \to \infty} (P^{\mu})^t(s,s')$ for any two states $s, s' \in \mathcal{S}$. *Note that we use π to denote the stationary distribution and μ to denote the policy.*

We next provide definitions of two norms that we will have occasion to use later. For a positive definite matrix $A \in \mathbb{R}^{n \times n}$, we define the inner product $\langle x, y \rangle_A = x^T A y$ and the associated norm $\|x\|_A = \sqrt{x^T A x}$ respectively. Since the numbers π_s are positive for all $s \in \mathcal{S}$, then the diagonal matrix $D = \text{diag}(\pi_1, \cdots, \pi_n) \in \mathbb{R}^{n \times n}$ is positive definite. Therefore, for any two vectors $V, V' \in \mathbb{R}^n$, we can also define an inner product as $\langle V, V' \rangle_D = V^T D V' = \sum_{s \in \mathcal{S}} \pi_s V(s) V'(s)$,and the associated norm as

$$\|V\|_D^2 = V^T D V = \sum_{s \in \mathcal{S}} \pi_s V(s)^2.$$

Finally, we introduce the definition of Dirichlet seminorm, following the notation of [14]:

$$\|V\|_{\text{Dir}}^2 = \frac{1}{2} \sum_{s,s' \in \mathcal{S}} \pi_s P^{\mu}(s,s') (V(s') - V(s))^2.$$

B. Temporal Difference Learning

We next introduce the update rule of the classical temporal difference method with linear function approximation V_{θ}, a linear function of θ:

$$V_{\theta}(s) = \sum_{l=1}^{K} \theta_l \phi_l(s) \quad \forall s \in \mathcal{S}, \tag{1}$$

where $\phi_l = (\phi_l(1), \cdots, \phi_l(n))^T \in \mathbb{R}^n$ for $l \in [K]$ are K given feature vectors. Together, all K feature vectors form a $n \times K$ matrix $\Phi = (\phi_1, \cdots, \phi_K)$. For $s \in \mathcal{S}$, let $\phi(s) = (\phi_1(s), \cdots, \phi_K(s))^T \in \mathbb{R}^K$ denote the s-th row of matrix Φ, a vector that collects the features of state s. Then, Eq. (1) can be written in a compact form $V_{\theta}(s) = \theta^T \phi(s)$. For brevity, we will omit the superscript μ throughout from now on.

The TD(0) method maintains a parameter $\theta(t)$ which is updated at every step to improve the approximation. Supposing that we observe a sequence of states $\{s(t)\}_{t \in \mathbb{N}_0}$, then the classical TD(0) algorithm updates as

$$\theta(t+1) = \theta(t) + \alpha_t \delta(t) \phi(s(t)),$$

where $\{\alpha_t\}_{t \in \mathbb{N}_0}$ is the sequence of step-sizes, and letting $s'(t)$ denote the next state after $s(t)$, the quantity $\delta(t)$ is the temporal difference error

$$\delta(t) = r(t) + \gamma \theta^T(t) \phi(s'(t)) - \theta^T(t) \phi(s(t)).$$

A common assumption on feature vectors in the literature [20], [2] is that features are linearly independent and uniformly bounded, which is formally given next.

Assumption 2. *The matrix Φ has full column rank, i.e., the feature vectors $\{\phi_1, \ldots, \phi_K\}$ are linearly independent. Additionally, we have that $\|\phi(s)\|_2^2 \leq 1$ for $s \in \mathcal{S}$.*

Under Assumption 1 and 2, we introduce the steady-state feature covariance matrix $\Phi^T D \Phi$. Note that, this is a positive definite matrix as an immediate consequence of Assumptions 1 and 2, and we let $\omega > 0$ be a lower bound on its smallest eigenvalue.

We will use the fact, shown in [20], that under Assumptions 1-2 as well as an additional assumption on the decay of the step-sizes α_t, the sequence of iterates $\{\theta_t\}$ generated by TD(0) learning converges almost surely to a vector satisfying a certain projected Bellman equation; we will use θ^* to refer to this vector.

C. The Distributed Model

We consider the scenario where each agent has its own independently evolving copy of the same MDP. More formally, each agent has the same 6-tuple $(\mathcal{S}, \mathcal{V}, \mathcal{A}, \mathcal{P}, r, \gamma)$; at time t, agent v will be in a state $s_v(t)$; it will apply action $a_v(t) \in \mathcal{A}$ with probability $\mu(a_v(t)|s_v(t))$; then agent v moves to state $s_v'(t)$ with probability $\mathcal{P}(s_v'(t)|s_v(t), a_v(t))$, with the transitions of all agents being independent of each other; finally agent v gets a reward $r_v(t) = r(s_v(t), a_v(t), s_v'(t))$. Note that, although the rewards obtained by different agents can be different, the reward function $r(s,a,s')$ is identical across agents.

Naturally, each agent can easily compute θ^* by simply ignoring all the other agents and running TD(0) locally. However, this ignores the possibility that agents can benefit from communication with each other. Along these lines, we propose our main method below as Algorithm 1: each agent runs TD(0) locally without any communication, and, at the end, the agents simply average the results.

Algorithm 1 Parallel TD(0)

```
1: For $v \in \mathcal{V}$, initialize $\theta_v(0)$, $s_v(0)$
2: for $t = 0$ to $T-1$ do
3:     for $v \in \mathcal{V}$ do
4:         Observe a tuple $(s_v(t), s'_v(t), r_v(t))$.
5:         Compute temporal difference:
               $\delta_v(t) = r_v(t) - \left(\phi(s_v(t)) - \gamma\phi(s'_v(t))\right)^T \theta_v(t).$   (2)
6:         Execute local TD update:
               $\theta_v(t+1) = \theta_v(t) + \alpha_t \delta_v(t) \phi_v(s_v(t)).$   (3)
7:         Update the local running average:
               $\hat{\theta}_v(t+1) = \left(1 - \frac{1}{t+2}\right)\hat{\theta}_v(t) + \frac{1}{t+2}\theta_v(t+1).$
8:     end for
9: end for
10: Return $\hat{\theta}(T) = \frac{1}{N}\sum_{v\in\mathcal{V}} \hat{\theta}_v(T)$ and $\bar{\theta}(T) = \frac{1}{N}\sum_{v\in\mathcal{V}} \theta_v(T)$.
```

Distributed implementation: The final averaging step represents the only interactions among the agents. Under the assumption that the nodes are connected to a server, computing the average in step 10 takes a single round of communication with a server. In the more common "nearest neighbor" model where the agents are connected over an undirected graph and nodes know the total number of nodes N, it is possible to find an ε-approximation of the average in $O(N\log(1/\varepsilon))$ time using the average consensus algorithm from [15]. One could also a finite-time average consensus method, see e.g., [6]. If knowledge of the number of nodes not available, and the communication graph is further time-varying, it is possible to do the same in $O(N^2\log(1/\varepsilon)$ using the average consensus algorithm from [13]. Finally, if the underlying graph is directed, one can use the popular push-sum for average consensus method [7], [1] whose convergence rate is geometric, though the question of whether a version of it can have a polynomial convergence rate in terms of N is open.

As we will later discuss, it suffices to choose ε in the previous paragraph proportional to a power of $1/T$ (where T is the number of iterations, decided on ahead of time), so that *the distributed message complexity of step 10 is $O(\log T)$ under any of the models discussed.*

III. Convergence Analyses of Our Method

We next describe the main result of this paper, which is an analysis of Algorithm 1 under the assumption that the tuples in step 4 are i.i.d. In the literature, the i.i.d model is sometimes referred to as having a "generator" for the MDP and is a more restrictive assumption compared to assuming that the state evolves as a Markov process with a fixed starting state. Nevertheless, this is a standard assumption under which many TD and Q-learning methods are analyzed (e.g., [3], [4], [5], [9] among others).

We begin with some notations. For centralized TD(0), convergence bounds generally scale both with the distance to the initial solution, and with the variance of the temporal difference error with average reward:

$$\sigma^2 = \mathrm{E}\left[\left(r(s,a,s') - \left(\phi(s) - \gamma\phi(s')\right)^T\theta^*\right)^2\right].$$

Here the expectation is taken with respect to the distribution that generates the state s with probability π_s, the actions $(a_1,\ldots,a_n)$ from the policy, and the next state $s'(t)$ from the transition of the MDP. We will use the same notation in the distributed setting, where this quantity is identical across agents, since the agents are all simulating the same MDP.

Further, we need a notion of the initial distance to the optimal solution; for simplicity, we take the maximum over all the agents to define:

$$\hat{R}_0 = \max_{v\in\mathcal{V}} E\left[\|\theta_v(0) - \theta^*\|_2^2\right].$$

The following theorem is our main result. Note that the equations are color-coded, with the meaning of the colors explained below.

Theorem 1. *Suppose Assumptions 1-2 hold and suppose that the tuples in step 4 of Algorithm 1 are generated i.i.d. with each $s_v(t)$ sampled from the stationary distribution π, and $r_v(t)$ being the reward and $s'_v(t)$ being the next state when the action is taken from the policy μ. Then:*

(a) For any constant step-size sequence $\alpha_0 = \cdots = \alpha_T = \alpha \le (1-\gamma)/8$, we have

$$E\left[(1-\gamma)\left\|V_{\theta^*} - V_{\hat{\theta}(T)}\right\|_D^2 + \gamma\left\|V_{\theta^*} - V_{\hat{\theta}(T)}\right\|_{\mathrm{Dir}}^2\right] \le \frac{1}{T}\left(\frac{1}{2\alpha}E\left[\|\bar{\theta}(0) - \theta^*\|_2^2\right] + \frac{4\hat{R}_0}{1-\gamma}\right) + \frac{\alpha\sigma^2}{N} + \frac{8\alpha^2\sigma^2}{1-\gamma}.$$

(b) For any $T \ge \frac{64}{(1-\gamma)^2}$ and constant step-size sequence $\alpha_0 = \cdots = \alpha_T = \frac{1}{\sqrt{T}}$, we have

$$E\left[(1-\gamma)\left\|V_{\theta^*} - V_{\hat{\theta}(T)}\right\|_D^2 + \gamma\left\|V_{\theta^*} - V_{\hat{\theta}(T)}\right\|_{\mathrm{Dir}}^2\right] \le \frac{1}{2\sqrt{T}}\left(E\left[\|\bar{\theta}(0) - \theta^*\|_2^2\right] + \frac{2\sigma^2}{N}\right) + \frac{1}{T}\left(\frac{4\hat{R}_0 + 8\sigma^2}{1-\gamma}\right).$$

(c) For the decaying step-size sequence $\alpha_t = \frac{\alpha}{t+\tau}$ with $\alpha = \frac{2}{(1-\gamma)\omega}$ and $\tau = \frac{16}{(1-\gamma)^2\omega}$. Then,

$$E\left[\|\bar{\theta}(t+1) - \theta^*\|_2^2\right] \le \frac{2\alpha^2\sigma^2/N}{t+\tau} + \frac{8\alpha^2\hat{\zeta}}{(t+\tau)^2} + \frac{(\tau-1)^4 E\left[\|\bar{\theta}(0) - \theta^*\|_2^2\right]}{(t+\tau)^4},$$

where $\hat{\zeta} = \max\left\{2\alpha^2\sigma^2, \tau\hat{R}_0\right\}$.

The proof of Theorem 1 is given in the section IV. To parse Theorem 1, note that all the terms in brown are "negligible" in a limiting sense. Indeed, in part (a), the first brown term scales as $O(1/T)$ and consequently goes to zero as $T \to \infty$ (whereas the remaining terms do not). In parts (b) and (c), the terms in brown go to zero at an asymptotically faster rate compared to the dominant term (i.e., as $1/T$ vs the dominant $1/\sqrt{T}$ term in part(b) and as $1/t^2, 1/t^4$ compared to the dominant $1/t$ in part (c)). Finally, the last term in part (a) scales as $O(\alpha^2)$ and will be negligible compared to the term preceding it, which scales as $O(\alpha)$, when α is small.

Moreover, among the non-negligible terms, whenever σ^2 appears, it is divided by N; this is highlighted in blue.

To summarize: parts (b) and (c) show that, when the number of iterations is large enough, we can divide the variance term by N as a consequence of the parallelism among N agents. Part (a) shows that, when the number of iterations is large enough and the step-size is small enough, the size of the final error will be divided by N.

Note that, in part (c), the result of this is a factor of N speed up of the entire convergence time (when T is large enough). In part (a), this results in a factor of N shrinking of the asymptotic error (when the step-size α is small enough). In part (b), however, this only shrinks the "variance term" by a factor of N; the term depending on the initial condition is unaffected. The explanation for this is that in parts (a) and (c), the variance of the temporal difference error dominates the convergence rate, while in part (b) this is not the case.

As far as we are aware, these results constitute the first example where parallelism was shown to be helpful for distributed temporal difference learning.

Required accuracy for the averaging step. For simplicity, we have given Theorem 1 under the assumption that the final averages $\hat{\theta}(T), \bar{\theta}(T)$ are computed exactly. We now come back to the question of how accurate the final averaging step needs to be to preserve our theoretical guarantees. It is immediate that all the quantities we bound in Theorem 1 (i.e., the left-hand sides of all the equations) are Lipschitz in a neighborhood of θ^*. Thus in Theorem 1(a) we need only a constant error in the averaging step, while in Theorem 1(b) and 1(c) we need an error rate proportional to a power or $1/T$. Since all average consensus methods previously discussed compute an ε-approximation to average consensus in $O(\log 1/\varepsilon)$ steps (treating all other variables as constants), *this means that step 10 in our method requires us to run a distributed average consensus method for at most $O(\log T)$ (treating all variables except T as constants) as previously claimed.*

IV. Proof of our main result

We now provide the proof of Theorem 1. Let $\Theta(t) \in \mathbb{R}^{N \times K}$ be a matrix whose rows are $\theta_1^T(t), \cdots, \theta_N^T(t)$. The following proposition follows immediately from the definitions (and recall here our notation of putting a bar to denote the network-wide average).

Proposition 1. *Suppose Assumptions 1-2 hold, and suppose that $\{\theta_v(t)\}_{v \in \mathcal{V}}$ are generated by Algorithm 1. Then,*

(a) $\bar{h}(t)$ is a linear function of $\bar{\theta}(t)$ and we can write $\bar{h}(t) = b - A\bar{\theta}(t)$.

(b) The conditional expectation of $\bar{m}(t)$ given $\Theta(t)$ is equal to zero:

$$E[\bar{m}(t) | \Theta(t)] = 0. \tag{4}$$

Our next step is to prove a recurrence relation satisfied by the average of the iterates, stated as the following lemma. Recall that θ^* is the fixed point of TD(0) on the MDP $(\mathcal{S}, \mathcal{V}, \mathcal{A}, \mathcal{P}, r, \gamma)$.

Lemma 1. *Suppose Assumptions 1-2 hold. Further suppose that $\{\theta_v\}_{v \in \mathcal{V}}$ are generated by Algorithm 1. For $t \in \mathbb{N}_0$, we have that*

$$\begin{aligned} E\left[\|\bar{\theta}(t+1) - \theta^*\|_2^2\right] &\leq E\left[\|\bar{\theta}(t) - \theta^*\|_2^2\right] \\ &+ \alpha_t^2 \left(\frac{2\sigma^2}{N} + \frac{8}{N} \sum_{v \in \mathcal{V}} E\left[\|V_{\theta_v(t)} - V_{\theta^*}\|_D^2\right] \right) \\ &- 2\alpha_t E\left[(1-\gamma) \left\| V_{\theta^*} - V_{\bar{\theta}(t)} \right\|_D^2 + \gamma \left\| V_{\theta^*} - V_{\bar{\theta}(t)} \right\|_{\text{Dir}}^2 \right]. \end{aligned} \tag{5}$$

Proof of Lemma 1. We have $\bar{\theta}(t+1) = \bar{\theta}(t) + \alpha_t \left[\bar{h}(t) + \bar{m}(t)\right]$. By taking expectations:

$$\begin{aligned} &E\left[\|\bar{\theta}(t+1) - \theta^*\|_2^2\right] = E\left[\|\bar{\theta}(t) - \theta^*\|_2^2\right] + \alpha_t^2 E\left[\|\bar{h}(t) + \bar{m}(t)\|_2^2\right] \\ &- 2\alpha_t E\left[\left(\bar{h}(t) + \bar{m}(t)\right)^T \left(\theta^* - \bar{\theta}(t)\right)\right]. \end{aligned} \tag{6}$$

We first consider the second term on the right hand side of Eq. (6). Following the definition of $\bar{h}(t)$ and $\bar{m}(t)$ and plugging in the expression for TD error $\delta_v(t)$ with Eq. (2), we obtain $E\left[\left\|\bar{h}(t) + \bar{m}(t)\right\|_2^2\right] = E\left[\|a^* - b^*\|_2^2\right]$, where

$$\boldsymbol{a}^* = \frac{1}{N} \sum_{v \in \mathcal{V}} \left[r_v(t) - \left(\phi(s_v(t)) - \gamma\phi(s_v'(t))\right)^T \theta^* \right] \phi(s_v(t)),$$

$$\boldsymbol{b}^* = \frac{1}{N} \sum_{v \in \mathcal{V}} \phi(s_v(t)) \left(\phi(s_v(t)) - \gamma\phi(s_v'(t))\right)^T \left(\theta_v(t) - \theta^*\right).$$

Using inequality $\|\boldsymbol{a}^* - \boldsymbol{b}^*\|^2 \leq 2\|\boldsymbol{a}^*\|^2 + 2\|\boldsymbol{b}^*\|^2$, we obtain

$$\begin{aligned} E\left[\left\|\bar{h}(t) + \bar{m}(t)\right\|_2^2\right] \leq& 2E\left[\|\boldsymbol{a}^*\|^2\right] + 2E\left[\|\boldsymbol{b}^*\|^2\right] \\ \leq& \frac{2\sigma^2}{N} + \frac{8}{N} \sum_{v \in \mathcal{V}} E\left[\|V_{\theta_v(t)} - V_{\theta^*}\|_D^2\right]. \end{aligned} \tag{7}$$

We next consider the third term on the right hand side of Eq. (6):

$$\begin{aligned} &E\left[\left[\bar{h}(t) + \bar{m}(t)\right]^T \left(\theta^* - \bar{\theta}(t)\right)\right] = E\left[\bar{h}^T(t)(\theta^* - \bar{\theta}(t))\right] \\ =& E\left[(1-\gamma) \left\| V_{\theta^*} - V_{\bar{\theta}(t)} \right\|_D^2 + \gamma \left\| V_{\theta^*} - V_{\bar{\theta}(t)} \right\|_{\text{Dir}}^2 \right]. \end{aligned} \tag{8}$$

Here we use that by Proposition 1 part (a), we have that $\bar{h}(t) = b - A\bar{\theta}(t)$. Furthermore, if we let $\bar{h}(\theta)$ denote the linear function $b - A\theta$, we have that $\bar{h}(\theta^*) = 0$. Now applying Corollary 1 in [11] proves the last equation.

Combining equations (6), (7), and (8), we obtain Eq.(5) ■

With this lemma in place, we are now ready to provide a proof of Theorem 1.

Proof of Theorem 1. Starting from Lemma 1 and Eq.(5), we first consider the bound for the term $\sum_{t=1}^T \sum_{v=1}^N E\left[\|V_{\theta_v(t)} - V_{\theta^*}\|_D^2\right]$. We can plug in that $N = 1$ into Lemma 1 to obtain the next inequality:

$$E\left[\|\theta_v(t+1) - \theta^*\|_2^2\right] \leq E\left[\|\theta_v(t) - \theta^*\|_2^2\right]$$

$$
\begin{aligned}
&+\alpha_t^2\left(2\sigma^2+8E\left[\|V_{\theta_v(t)}-V_{\theta^*}\|_D^2\right]\right)\\
&-2\alpha_t E\left[(1-\gamma)\left\|V_{\theta^*}-V_{\theta_v(t)}\right\|_D^2+\gamma\left\|V_{\theta^*}-V_{\theta_v(t)}\right\|_{\mathrm{Dir}}^2\right].
\end{aligned}
$$

If the sequence of step-sizes are non-increasing and satisfies $8\alpha_t^2-2\alpha_t(1-\gamma)\le-\alpha_t(1-\gamma)$, then we obtain

$$
\begin{aligned}
&\alpha_t E\left[(1-\gamma)\|V_{\theta^*}-V_{\theta_v(t)}\|_D^2+2\gamma\|V_{\theta^*}-V_{\theta_v(t)}\|_{\mathrm{Dir}}^2\right]\\
\le&E\left[\|\theta_v(t)-\theta^*\|_2^2\right]-E\left[\|\theta_v(t+1)-\theta^*\|_2^2\right]+2\alpha_t^2\sigma^2.
\end{aligned}
$$

Since $E\left[2\gamma\|V_{\theta^*}-V_{\theta_v(t)}\|_{\mathrm{Dir}}^2\right]$ is non-negative, it now follows that

$$
\begin{aligned}
&\alpha_t E\left[(1-\gamma)\|V_{\theta^*}-V_{\theta_v(t)}\|_D^2\right]\\
\le&E\left[\|\theta_v(t)-\theta^*\|_2^2\right]-E\left[\|\theta_v(t+1)-\theta^*\|_2^2\right]+2\alpha_t^2\sigma^2.
\end{aligned}
$$

Multiplying α_t on both sides and summing over t, we have

$$
\begin{aligned}
&\sum_{t=0}^{T-1}\alpha_t^2E\left[(1-\gamma)\left\|V_{\theta^*}-V_{\theta_v(t)}\right\|_D^2\right]\\
=&\alpha_0E\left[\|\theta_v(0)-\theta^*\|_2^2\right]+\sum_{t=1}^{T-1}(\alpha_{t-1}-\alpha_t)E\left[\|\theta_v(t)-\theta^*\|_2^2\right]\\
&-\alpha_{T-1}E\left[\|\theta_v(T)-\theta^*\|_2^2\right]+2\sum_{t=0}^{T-1}\alpha_t^3\sigma^2\\
\le&\alpha_0E\left[\|\theta_v(0)-\theta^*\|_2^2\right]+2\sum_{t=0}^{T-1}\alpha_t^3\sigma^2,
\end{aligned}
$$

where the last inequality is because that $\{\alpha_t\}_t$ are non-increasing step-sizes. Summing over agents v, we get

$$
\sum_{v=1}^{N}\sum_{t=0}^{T-1}\alpha_t^2E\left[(1-\gamma)\left\|V_{\theta^*}-V_{\theta_v(t)}\right\|_D^2\right]\le N\alpha_0\hat{R}_0+2N\sum_{t=0}^{T-1}\alpha_t^3\sigma^2. \tag{9}
$$

With this equation in place, we now turn to the proof of all the parts of the theorem.

Proof of part (a): We consider the constant step-size sequence $\alpha_0=\cdots=\alpha_T\le(1-\gamma)/8$. Then let α denote the constant step-size. Plugging into Eq. (5) and rearranging it, we get

$$
\begin{aligned}
&2\alpha E\left[(1-\gamma)\left\|V_{\theta^*}-V_{\bar{\theta}(t)}\right\|_D^2+\gamma\left\|V_{\theta^*}-V_{\bar{\theta}(t)}\right\|_{\mathrm{Dir}}^2\right]\\
\le&E\left[\left\|\bar{\theta}(t)-\theta^*\right\|_2^2\right]-E\left[\left\|\bar{\theta}(t+1)-\theta^*\right\|_2^2\right]\\
&+\alpha^2\left(\frac{2\sigma^2}{N}+\frac{8}{N}\sum_{v\in\mathcal{V}}E\left[\|V_{\theta_v(t)}-V_{\theta^*}\|_D^2\right]\right).
\end{aligned}
$$

Summing over t gives

$$
\begin{aligned}
&2\sum_{t=0}^{T-1}\alpha E\left[(1-\gamma)\left\|V_{\theta^*}-V_{\bar{\theta}(t)}\right\|_D^2+\gamma\left\|V_{\theta^*}-V_{\bar{\theta}(t)}\right\|_{\mathrm{Dir}}^2\right]\\
\le&E\left[\left\|\bar{\theta}(0)-\theta^*\right\|_2^2\right]-E\left[\left\|\bar{\theta}(T)-\theta^*\right\|_2^2\right]\\
&+\frac{2T\alpha^2\sigma^2}{N}+\frac{8}{N}\sum_{t=0}^{T-1}\sum_{v\in\mathcal{V}}\alpha^2E\left[\|V_{\theta_v(t)}-V_{\theta^*}\|_D^2\right]\\
\le&E\left[\left\|\bar{\theta}(0)-\theta^*\right\|_2^2\right]+\frac{2T\alpha^2\sigma^2}{N}+\frac{8\alpha}{1-\gamma}\left(\hat{R}_0+2T\alpha^2\sigma^2\right)
\end{aligned}
$$

where we used Eq. (9).

Now dividing by 2α on both sides:

$$
\begin{aligned}
&\sum_{t=0}^{T-1}E\left[(1-\gamma)\left\|V_{\theta^*}-V_{\bar{\theta}(t)}\right\|_D^2+\gamma\left\|V_{\theta^*}-V_{\bar{\theta}(t)}\right\|_{\mathrm{Dir}}^2\right]\\
\le&\frac{1}{2\alpha}E\left[\left\|\bar{\theta}(0)-\theta^*\right\|_2^2\right]+\frac{T\alpha\sigma^2}{N}+\frac{4}{1-\gamma}\left(\hat{R}_0+2T\alpha^2\sigma^2\right).
\end{aligned}
$$

Let $\hat{\theta}(T)=\frac{1}{T}\sum_{t=1}^{T}\bar{\theta}(t)$. Then, by convexity

$$
\begin{aligned}
&E\left[(1-\gamma)\left\|V_{\theta^*}-V_{\hat{\theta}(T)}\right\|_D^2+\gamma\left\|V_{\theta^*}-V_{\hat{\theta}(T)}\right\|_{\mathrm{Dir}}^2\right]\\
\le&\frac{1}{T}\sum_{t=1}^{T}E\left[(1-\gamma)\left\|V_{\theta^*}-V_{\bar{\theta}(t)}\right\|_D^2+\gamma\left\|V_{\theta^*}-V_{\bar{\theta}(t)}\right\|_{\mathrm{Dir}}^2\right]\\
\le&\frac{1}{T}\left(\frac{1}{2\alpha}E\left[\left\|\bar{\theta}(0)-\theta^*\right\|_2^2\right]+\frac{4\hat{R}_0}{1-\gamma}\right)+\frac{\alpha\sigma^2}{N}+\frac{8\alpha^2\sigma^2}{1-\gamma},
\end{aligned}
$$

which is what we wanted to show.

Proof of part (b): We now consider the step-size $\alpha_0=\cdots=\alpha_T=\frac{1}{\sqrt{T}}$. When $T\ge\frac{64}{(1-\gamma)^2}$, it can be observed that $\alpha=\frac{1}{\sqrt{T}}\le\frac{1-\gamma}{8}$. As a consequence of part (a), it is immediate that,

$$
\begin{aligned}
&E\left[(1-\gamma)\left\|V_{\theta^*}-V_{\hat{\theta}(T)}\right\|_D^2+\gamma\left\|V_{\theta^*}-V_{\hat{\theta}(T)}\right\|_{\mathrm{Dir}}^2\right]\\
\le&\frac{1}{2\sqrt{T}}\left(E\left[\left\|\bar{\theta}(0)-\theta^*\right\|_2^2\right]+\frac{2\sigma^2}{N}\right)+\frac{1}{T}\left(\frac{4\hat{R}_0+8\sigma^2}{1-\gamma}\right),
\end{aligned}
$$

which is what we wanted to show.

Proof of part (c): Using that $\gamma\left\|V_{\theta^*}-V_{\bar{\theta}(t)}\right\|_{\mathrm{Dir}}^2$ is non-negative and rearranging Eq. (5), we have

$$
\begin{aligned}
&E\left[\left\|\bar{\theta}(t+1)-\theta^*\right\|_2^2\right]\le\alpha_t^2\left(\frac{2\sigma^2}{N}+\frac{8}{N}\sum_{v\in\mathcal{V}}E\left[\|V_{\theta_v(t)}-V_{\theta^*}\|_D^2\right]\right)\\
&+E\left[\left\|\bar{\theta}(t)-\theta^*\right\|_2^2\right]-2\alpha_t(1-\gamma)E\left\|V_{\theta^*}-V_{\bar{\theta}(t)}\right\|_D^2.
\end{aligned}
$$

Applying Lemma 1 in [2], which states that $\sqrt{\omega}\|\theta\|_2\le\|V_\theta\|_D\le\|\theta\|_2$, we get

$$
\begin{aligned}
&E\left[\left\|\bar{\theta}(t+1)-\theta^*\right\|_2^2\right]\le(1-2\alpha_t(1-\gamma)\omega)E\left[\left\|\bar{\theta}(t)-\theta^*\right\|_2^2\right]\\
&+\alpha_t^2\left(\frac{2\sigma^2}{N}+\frac{8}{N}\sum_{v\in\mathcal{V}}E\left[\|V_{\theta_v(t)}-V_{\theta^*}\|_D^2\right]\right).
\end{aligned}
\tag{10}
$$

We first consider the last term on the right hand side, i.e., $E\left[\|V_{\theta_v(t)}-V_{\theta^*}\|_D^2\right]$. Since each agent in the system executes the classical TD(0) at time t for $t\in\mathbb{N}_0$, then by part (c) of Theorem 2 and Lemma 1 in [2], for $v\in\mathcal{V}$, we have that $E\left[\|V_{\theta_v(t)}-V_{\theta^*}\|_D^2\right]\le E\left[\|\theta_v(t)-\theta^*\|_2^2\right]\le\frac{\hat{\zeta}}{t+\tau}$, where $\hat{\zeta}=\max\left\{2\alpha^2\sigma^2,\tau\hat{R}_0\right\}$. Hence, $\frac{8}{N}\sum_{v\in\mathcal{V}}E\left[\|V_{\theta_v(t)}-V_{\theta^*}\|_D^2\right]\le\frac{8\hat{\zeta}}{t+\tau}$, and plugging it into Eq. (10), we can obtain

$$
\begin{aligned}
&E\left[\left\|\bar{\theta}(t+1)-\theta^*\right\|_2^2\right]\\
\le&(1-2\alpha_t(1-\gamma)\omega)E\left[\left\|\bar{\theta}(t)-\theta^*\right\|_2^2\right]+\alpha_t^2\left(\frac{2\sigma^2}{N}+\frac{8\hat{\zeta}}{t+\tau}\right)
\end{aligned}
$$

$$= \left(1 - \frac{4}{t+\tau}\right) E\left[\left\|\bar{\theta}(t) - \theta^*\right\|_2^2\right] + \frac{2\alpha^2\sigma^2/N}{(t+\tau)^2} + \frac{8\alpha^2\hat{\zeta}}{(t+\tau)^3},$$

where we use that $\alpha_t = \frac{\alpha}{t+\tau}$ with $\alpha = \frac{2}{(1-\gamma)\omega}$ and $\tau = \frac{16}{(1-\gamma)^2\omega}$ to get the last line. This recursion now immediately implies part(c) of the theorem using the standard estimate

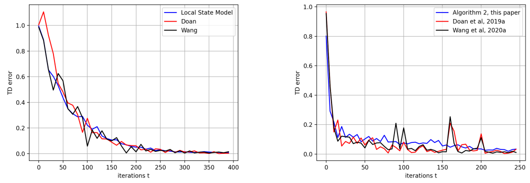
$$\prod_{i=0}^{t}\left(1 - \frac{4}{t+\tau-i}\right) < \left(\frac{\tau-1}{t+\tau}\right)^4. \tag{11}$$

■

V. Numerical Experiments

In this section, we perform some experiments comparing Algorithm 1 with earlier distributed TD methods from [5], [17] and [21] in terms of TD error. Note that the distributed TD methods of [5] and [17] are the same except that [5] has an additional projection step. We consider the case of constant step-size, which is the most widely used in practice, taking $N = 100$ agents. The communication graph among agents is generated by the Erdos–Renyi model, which is connected. We consider two simple examples: Gridworld (see Chapter 3 of [19]) and MountainCar-v1 from OpenAI Gym; for the latter, we use the tile coding [19] to discretize continuous state spaces into overlapping tiles. We use 5 tilings, and each tiling has 7×7 grids.

Recall that our method only uses one run of average consensus at the end, whereas the other methods require a communication at every step. The graphs for our method show the TD error at each iteration if we stopped the method and run the average consensus to average the estimates across the network. Figure 1 shows that the TD errors of Algorithm 1 perform essentially identically to the other methods in spite of the reduced communication.



(a) Grid World

(b) MountainCar-v1

Fig. 1: Comparison of our method to the previous literature for a policy that takes uniformly random actions.

VI. Conclusion

We have presented convergence results for distributed TD(0) with linear function approximation. Our results are unique in terms of utilizing almost no communication: only one run of average consensus is needed. In particular, this means we need to do $O(\log T)$ average consensus steps for T steps of TD(0) at every node of the network. The convergence bounds we derive reduce the variance by a factor of N when the nodes generate their samples independently. The main question left by this work is whether it is possible to extend these results to other methods popular in the reinforcement learning, such as Q-learning. It would also be of interest to to apply these results to policies in the context of control of epidemics using the problem formulation in [12].

References

[1] Florence Bénézit, Vincent Blondel, Patrick Thiran, John Tsitsiklis, and Martin Vetterli. Weighted gossip: Distributed averaging using non-doubly stochastic matrices. In *2010 ieee international symposium on information theory*, pages 1753–1757. IEEE, 2010.

[2] Jalaj Bhandari, Daniel Russo, and Raghav Singal. A finite time analysis of temporal difference learning with linear function approximation. In *Conference on Learning Theory*, pages 1691–1692, 2018.

[3] Gal Dalal, Balázs Szörényi, Gugan Thoppe, and Shie Mannor. Finite sample analyses for td (0) with function approximation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[4] Gal Dalal, Gugan Thoppe, Balázs Szörényi, and Shie Mannor. Finite sample analysis of two-timescale stochastic approximation with applications to reinforcement learning. In *Conference On Learning Theory*, pages 1199–1233. PMLR, 2018.

[5] Thinh Doan, Siva Maguluri, and Justin Romberg. Finite-time analysis of distributed td (0) with linear function approximation on multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 1626–1635, 2019.

[6] Mohammadreza Doostmohammadian. Single-bit consensus with finite-time convergence: Theory and applications. *IEEE Transactions on Aerospace and Electronic Systems*, 56(4):3332–3338, 2020.

[7] David Kempe, Alin Dobra, and Johannes Gehrke. Gossip-based computation of aggregate information. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 482–491. IEEE, 2003.

[8] Sajad Khodadadian, Pranay Sharma, Gauri Joshi, and Siva Theja Maguluri. Federated reinforcement learning: Linear speedup under markovian sampling. In *International Conference on Machine Learning*, pages 10997–11057. PMLR, 2022.

[9] Harshat Kumar, Alec Koppel, and Alejandro Ribeiro. On the sample complexity of actor-critic method for reinforcement learning with function approximation. *arXiv preprint arXiv:1910.08412*, 2019.

[10] Rui Liu and Alex Olshevsky. Distributed td (0) with almost no communication. *arXiv preprint arXiv:2104.07855*, 2021.

[11] Rui Liu and Alex Olshevsky. Temporal difference learning as gradient splitting. In *International Conference on Machine Learning*, pages 6905–6913. PMLR, 2021.

[12] Qianqian Ma, Yang-Yu Liu, and Alex Olshevsky. Optimal lockdown for pandemic control. *arXiv preprint arXiv:2010.12923*, 2020.

[13] Angelia Nedic, Alex Olshevsky, Asuman Ozdaglar, and John N Tsitsiklis. On distributed averaging algorithms and quantization effects. *IEEE Transactions on automatic control*, 54(11):2506–2517, 2009.

[14] Yann Ollivier. Approximate temporal difference learning is a gradient descent for reversible policies. *arXiv preprint arXiv:1805.00869*, 2018.

[15] Alex Olshevsky. Linear time average consensus and distributed optimization on fixed graphs. *SIAM Journal on Control and Optimization*, 55(6):3990–4014, 2017.

[16] Han Shen, Kaiqing Zhang, Mingyi Hong, and Tianyi Chen. Asynchronous advantage actor critic: Non-asymptotic analysis and linear speedup. *arXiv preprint arXiv:2012.15511*, 2020.

[17] Jun Sun, Gang Wang, Georgios B Giannakis, Qinmin Yang, and Zaiyue Yang. Finite-sample analysis of decentralized temporal-difference learning with linear function approximation. *In International Conference on Artificial Intelligence and Statistics*, pages 1–8, 2020.

[18] Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44, 1988.

[19] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT Press, 2018.

[20] John N Tsitsiklis and Benjamin Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5):674–690, 1997.

[21] Gang Wang, Songtao Lu, Georgios Giannakis, Gerald Tesauro, and Jian Sun. Decentralized td tracking with linear function approximation and its finite-time analysis. *Advances in Neural Information Processing Systems*, 33, 2020.