NUMERICAL EXPERIMENTS ON DISTRIBUTED TD(0) WITH LOCAL STATE

Ruizhao Zhu
Dept. of Electrical & Computer Engineering
Boston University
Boston, MA 02215
rzhu@bu.edu

Qianqian Ma
Dept.of Electrical & Computer Engineering
Boston University
Boston, MA 02215
maqq@bu.edu

Rui Liu
Division of Systems Engineering
Boston University
Boston, MA 02215
rliu@bu.edu

September 18, 2021

ABSTRACT

We study the distributed TD(0) with local state (Algorithm 2 in Liu and Olshevsky (2021)) and perform corresponding numerical experiments. The goal of experiments is to illustrate Theorem 2 in Liu and Olshevsky (2021), which shows a linear time speedup phenomenon. In particular, to the extent that the variance of the temporal difference error affects the performance of TD(0), the performance of distributed TD(0) with local state is a factor of N times better than the performance of regular TD(0), where N is the number of agents. We provide the results of numerical experiments on classic control problems in the OpenAI Gym and a grid world Markov Decision Process (MDP).

1 Distributed TD(0) with Local State

In this section, we first provide notation and standard background information on MDP and temporal difference learning with linear function approximation; then describe formally the distributed TD(0) with local state algorithm we analyzed and corresponding convergence results.

1.1 Markov Decision Processes

A discounted reward MDP is described by a 5-tuple $(\mathcal{S},\mathcal{A},\mathcal{P},r,\gamma)$, where $\mathcal{S}=\{1,2,\cdots,n\}$ is a finite state 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 $\mu(s,\cdot)$ over the action space $\mathcal{A}$, so that $\sum_{a\in\mathcal{A}}\mu(s,a)=1$. Fixing the policy μ induces a probability transition matrix between states:

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

We will use $r_{t+1}=r(s_t,a_t,s_{t+1})$ 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

$$V^{\mu}(s)=E_{\mu,s}\left[\sum_{t=0}^{\infty}\gamma^t r_{t+1}\right], \tag{1}$$

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} as a vector in $\mathbb{R}^n$ and treat P^{μ} as a matrix in $\mathbb{R}^{n\times n}$.

Next, we state a standard assumptions 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 $s, s' \in \mathcal{S}$.

We next provide definitions of two norms. For any two vectors $V, V' \in \mathbb{R}^n$, we 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:

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

1.2 Temporal Difference Learning

If the number of states is very large, it will be computationally expensive to evaluate the value function V^μ of a policy. Therefore, the classical TD algorithm uses low dimensional approximation V_θ^μ. For brevity, we will omit the superscript μ throughout from now on.

Here we study the simplest case where V_θ is a linear function of θ:

$$V_\theta(s) = \sum_{l=1}^{K} \theta_l \phi_l(s) \quad \forall s \in \mathcal{S}, \tag{2}$$

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 forms 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. (2) can be written in a compact form $V_\theta(s) = \theta^T \phi(s)$.

The method maintains a parameter $\theta(t)$ which is updated ate very 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:

$$\theta(t+1) = \theta(t) + \alpha_t \delta(t) \phi(s(t)), \tag{3}$$

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

$$\delta(t) = r(t) + \gamma \theta^T(t) \phi(s'(t)) - \theta^T(t) \phi(s(t)). \tag{4}$$

A common assumption on feature vectors 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, it satisfies that $\|\phi(s)\|_2^2 \le 1$ for $s \in \mathcal{S}$.*

Under Assumption 1 and 2, we introduce the steady-state feature covariance matrix $\Phi^T D \Phi$ and let $\omega > 0$ is a lower bound of the minimum eigenvalue of matrix $\Phi^T D \Phi$.

1.3 Distributed TD(0) with Local State and Its Convergence Results

We study the distributed model with local state, which is introduced in Liu and Olshevsky (2021). In the distributed model with local state, each agent has its own independently evolving copy of the same MDP. In particular, let $\mathcal{V} = \{1, \ldots, N\}$ denote the set of agents and each agent has the same 5-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$. 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 $\mu(s_v(t), a_v(t))$; 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.

We use the notation θ_{lc}^* to be the fixed point of TD(0) on the MDP $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$. Then, the distributed TD(0) in this setting, which is also called distributed TD(0) with local state, are formally given in the Algorithm 1.

Algorithm 1 TD(0) with Local State

For $v \in \mathcal{V}$, initialize $\theta_v(0)$, $s_v(0)$
for $t = 0$ to T **do**
  for $v \in \mathcal{V}$ **do**
    Observe a tuple $(s_v(t), s_v'(t), r_v(t))$.
    Compute temporal difference:

$$\delta_v(t) = r_v(t) - \left(\phi(s_v(t)) - \gamma\phi(s_v'(t))\right)^T \theta_v(t).$$

    Execute local TD update:

$$\theta_v(t+1) = \theta_v(t) + \alpha_t \delta_v(t)\phi_v(s(t)).$$

    Update running average:

$$\hat{\theta}_v(t+1) = \left(1 - \frac{1}{t+2}\right)\hat{\theta}_v(t) + \frac{1}{t+2}\theta_v(t+1).$$

  end for
end for
Return $\hat{\theta}(T)$ and $\bar{\theta}(T)$:

$$\hat{\theta}(T) = \frac{1}{N}\sum_{v\in\mathcal{V}} \hat{\theta}_v(T), \quad \bar{\theta}(T) = \frac{1}{N}\sum_{v\in\mathcal{V}} \theta_v(T).$$

Before stating convergence results, we first introduce some notation. For the local model, the variance of the temporal difference error is identical to the variance defined in the centralized model:

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

In the multi-agent case, we simply take the maximum of initial condition over all the agents to define:

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

The convergence result proved in Liu and Olshevsky (2021) is given in the subsequent theorem.

Theorem 1. *[Theorem 2 in Liu and Olshevsky (2021)] Suppose Assumptions 1-2 hold. Suppose further that* $\{\theta_v(t)\}_{v\in\mathcal{V}}$ *and* $\{\hat{\theta}_v(t)\}_{v\in\mathcal{V}}$ *are generated by Algorithm 1 in the local state model. Then,*

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

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

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

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

(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,*

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

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

To parse Theorem 1, note that all the terms in brown are "negligible" in a limiting sense. Indeed, in part (a), the first 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. We refer to this as the linear time speedup phenomenon.

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.

2 Experiments on Open AI Gym

In this section, we apply Algorithm 1 on Mountain Car and Cartpole in OpenAI Gym to show the linear time speedup phenomenon. We use TD(0) as baseline to compare with. We will introduce the feature we use, tile coding and the results for both situations.

2.1 Tile Coding

We are using tile coding Sutton and Barto (2018) as our feature for linear approximation. Tile code is a method for coarse coding. In tile coding the receptive fields of the features are grouped into exhaustive partitions of input space. Each such partition is called a tiling, and each element of the partition is called a tile. Each tile is a the receptive field for one binary feature.

2.2 MountainCar Setup

The goal for MountainCar problem is to find a policy to let the car reach the flag on the mountain, namely reach position at 0.5 in this problem. The observation space is a 2-dim continuous space, each represents car position and car velocity respectively. There are 3 actions, accelerate to the left, do not accelerate and accelerate to the right. The reward is 0 if the agent reaches the flag (position $= 0.5$) and the reward is -1 if the agent not reaches the flag (position $= 0.5$). For feature selection, we use 5 tilings each tile contain a 7×7 grids. Then, the feature dimension is $5 \times 7 \times 7$. The state space range is given from the definition of the problem. The car position is in the range $[-1.2, 0.6]$ and car velocity is in the range $[-0.07, 0.07]$. We fix the policy as uniform random policy selecting from the 3 possible actions. We only test the i.i.d. case described by the theorem, where at each iteration we get a random initialization of states in the given state space range.

2.3 CartPole Setup

The goal for CartPole problem is to find a policy to let the Pole maintain not falling as long as possible. The observation space is a 4-dim continuous space, each represents cart position, cart velocity respectively, pole angle and pole angle velocity. There are 2 actions, push cart to the left and push cart the right. The reward is 1 if the pole not falling. For feature selection, we use 5 tilings each tile contain a $7 \times 7 \times 7 \times 7$ grids. Then, the feature dimension is $5 \times 7 \times 7 \times 7 \times 7$. The state space range is given from the definition of the problem. The cart position is in the range $[-1.2, 0.6]$ and car velocity is in the range $[-0.07, 0.07]$. We fix the policy as uniform random policy selecting from the 3 possible actions. We only test the i.i.d. case described by the theorem, where at each iteration we get a random initialization of states in the given state space range.

2.4 Results

We have done experiments based on several metrics. We mainly use the metric $\left\|\bar{\theta}(t+1) - \bar{\theta}(t)\right\|_2^2$ and $\left\|\frac{\bar{\theta}(t+1)-\bar{\theta}(t)}{\alpha_t}\right\|_2^2$. They are the same for the constant step-size α_t, except the different scale. Therefore, we only show results of $\left\|\frac{\bar{\theta}(t+1)-\bar{\theta}(t)}{\alpha_t}\right\|_2^2$ for constant step-size. They are different for the decaying step-size α_t and we will show both in that case.

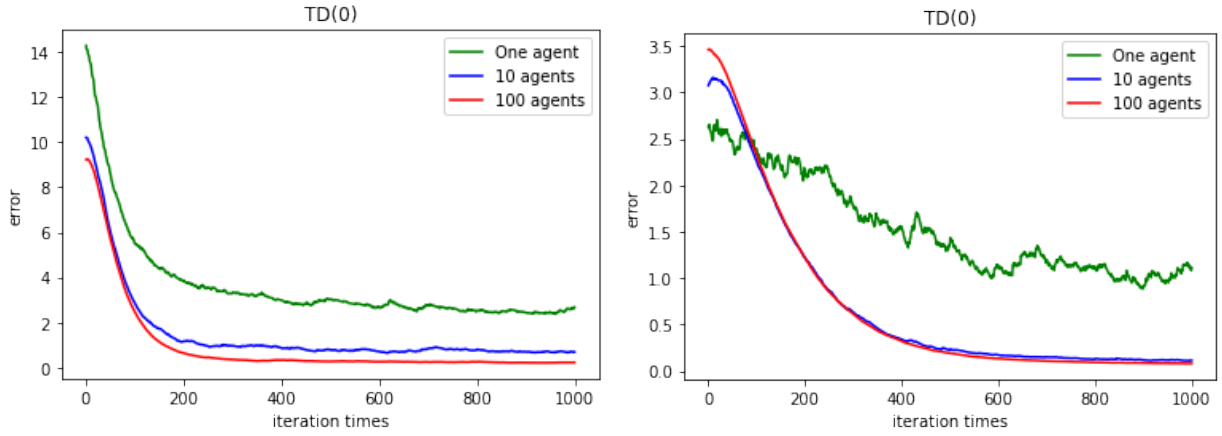
Following Eq.(32) in Liu and Olshevsky (2021), we have

$$\left\|\frac{\bar{\theta}(t+1) - \bar{\theta}(t)}{\alpha_t}\right\|_2^2 = \left\|\frac{1}{N}\sum_{v\in\mathcal{V}} \delta_v(t)\phi\left(s_v(t)\right)\right\|_2^2 \leq \frac{2\sigma^2}{N} + \frac{8}{N}\sum_{v\in\mathcal{V}} E\left[\left\|V_{\theta_v(t)} - V_{\theta^*_{\text{lc}}}\right\|_D^2\right]. \tag{5}$$

Then, the upper bounds of both metrics do not converge to zero for constant step-size, while the upper bound of $\left\|\bar{\theta}(t+1)-\bar{\theta}(t)\right\|_2^2$ converges to zero for decaying step-size when t goes to infinity.

2.4.1 Constant Step-size with i.i.d Observation

Figure 1 shows that, with constant step-size $\alpha_t = 0.01$, MountainCar and CartPole both get benefits from parallelism, although it is not a linear time speedup. It is because that only the upper bound has a linear time speedup following Eq.(5). It can be observed that, it is a significant improvement if we use 10 agents instead of one agent. However, it only makes small progress if we use 100 agents instead of 10 agents.



(a) (b)

Figure 1: CartPole and Mountain car with $\alpha_t = 0.01$, $\gamma = 0.8$. (a) CartPole, (b) MountainCar

The performance of different discount factor is of interest. Figure 2 shows CartPole results of different value of γ with constant step-size $\alpha_t = 0.01$. Regardless what value of γ we use, the algorithm accelerate when enlarging the number of agents. We can also see that the speed up is more obvious when γ is not very close to 1.

2.4.2 Decaying Step-size with i.i.d Observation

Figure 3 shows the $\left\|\frac{\bar{\theta}(t+1)-\bar{\theta}(t)}{\alpha_t}\right\|_2^2$ with decaying step-size. Figure 4 shows the $\left\|\bar{\theta}(t+1)-\bar{\theta}(t)\right\|_2^2$ with decaying step-size. We can see both $\frac{1}{t}$ and $\frac{1}{\sqrt{t}}$ cases are converging more quickly to zero as expected. However, the improvement got by parallelism with decaying step-size is smaller than the improvement with the constant step-size.

2.5 More Discussions

One thing we notice later is that in the code, the default initialization range of feature is smaller than the possible range in OpenAI gym. To generate i.i.d. observations, we reset the environment at each step. This causes we only update a small number of dimension in the parameter θ. We then modified the initialization range of feature to its largest range getting a result like in Figure 5 (a), the TD-update is much more flat. Figure 5 (b) shows the 1 agent case in 10000 iterations. It indeed converges but much slower.

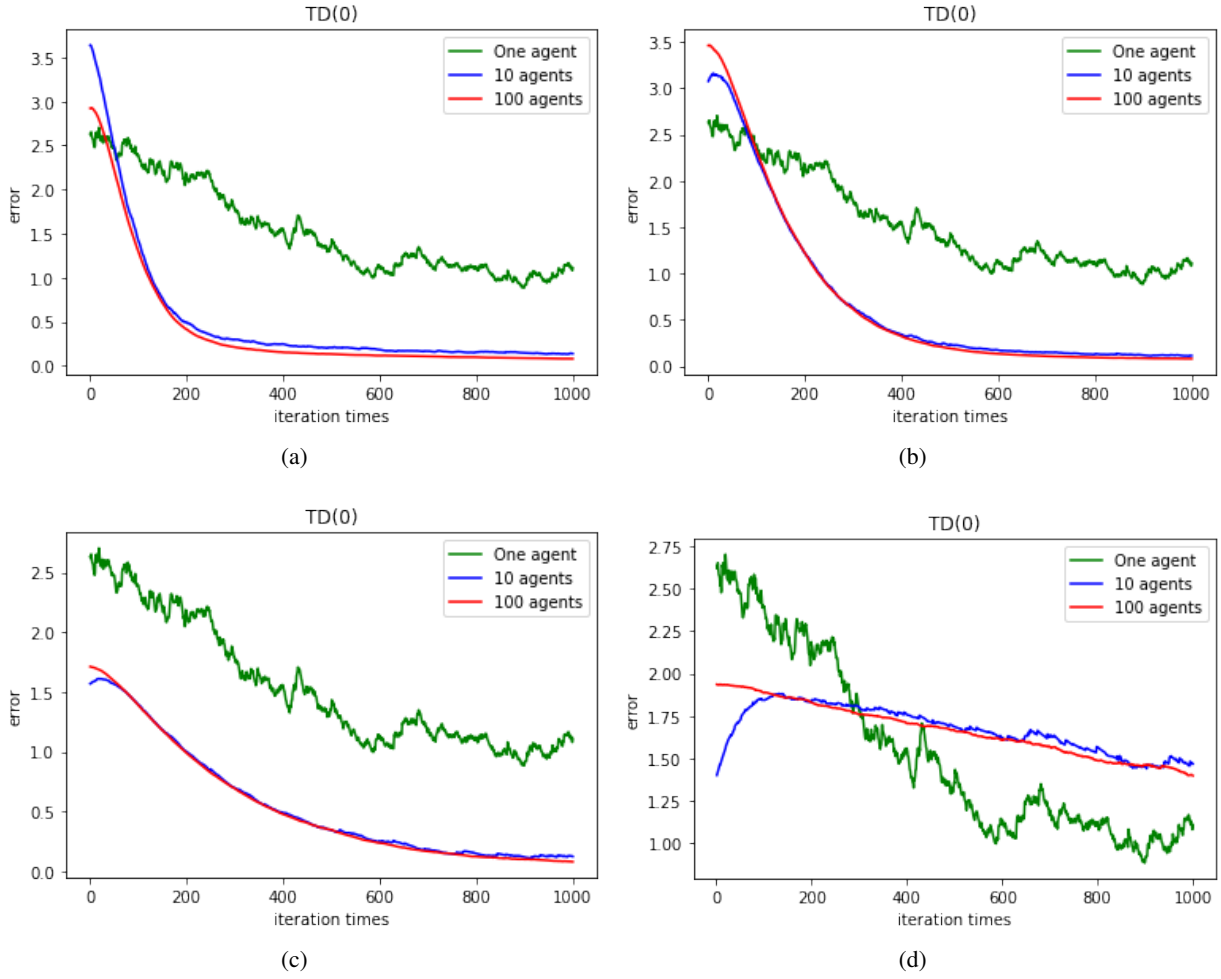


Figure 2: Mountain car with different γ. (a) $\gamma = 0.5$, (b) $\gamma = 0.8$, (c) $\gamma = 0.9$, (d) $\gamma = 0.99$

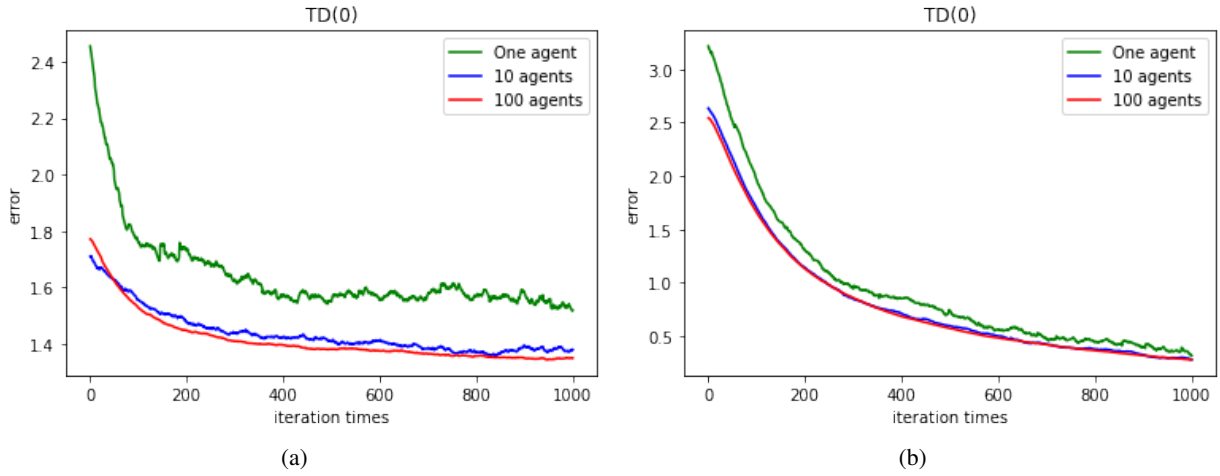
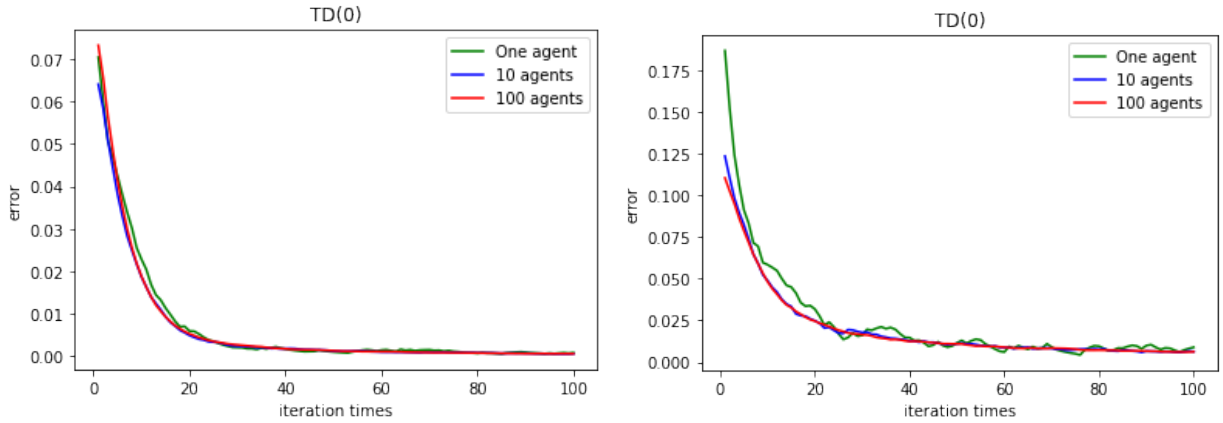
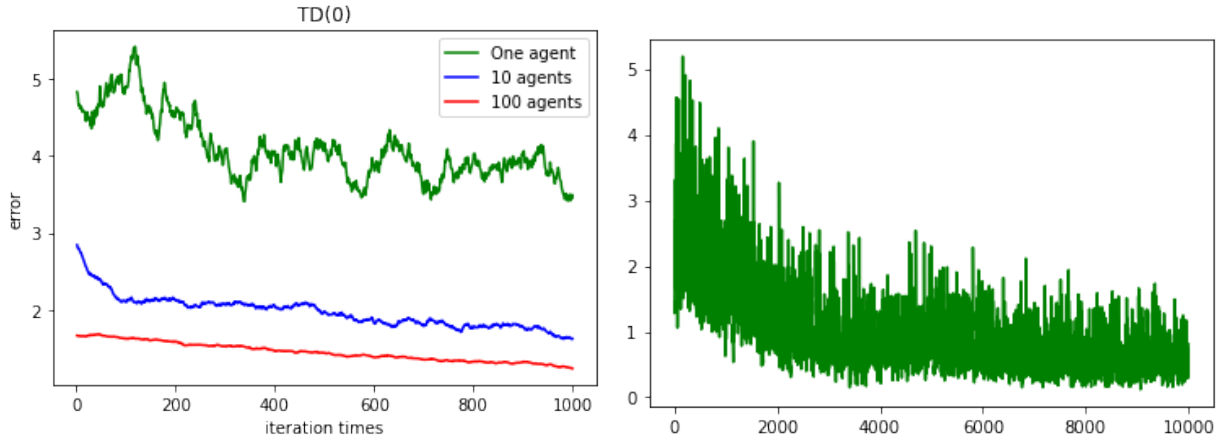


Figure 3: Mountain car with decaying step-size and discount factor $\gamma = 0.8$. We are using the metric $\left\| \frac{\bar{\theta}(t+1)-\bar{\theta}(t)}{\alpha_t} \right\|_2^2$ (a) $\alpha_t = \frac{1}{t}$, (b) $\alpha_t = \frac{1}{\sqrt{t}}$



(a) (b)

Figure 4: Mountain car with decaying step-size and discount factor $\gamma = 0.8$. We are using the metric $\left\|\bar{\theta}(t+1) - \bar{\theta}(t)\right\|_2^2$ (a) $\alpha_t = \frac{1}{t}$, (b) $\alpha_t = \frac{1}{\sqrt{t}}$



(a) (b)

Figure 5: Mountain car with decaying step-size and discount factor $\gamma = 0.8$. We are using the metric $\left\|\bar{\theta}(t+1) - \bar{\theta}(t)\right\|_2^2$ with $\alpha_t = \frac{1}{t}$. The initialization range of feature is set to the largest possible range. (a) using different number of agents for 1000 iterations. (b) using 1 agent for 10000 iterations.

3 Experiments on a Grid World MDP

To further verify the effectiveness of Algorithm 1, we apply Algorithm 1 on a simple grid-world experiment. We consider a 4×4 grid, where the non-terminal states are $\mathcal{S} = \{1, 2, \ldots, 15\}$, the terminal state is the upper-left grid (as shown in Figure 6). There are four possible actions for each non-terminal state, $\mathcal{A}$ = {left, right, up, down}. If the action leads out of the grid, then the next state will remain to be the current state. In this experiment, we will consider a random policy, i.e., each agent choose action from the 4 possible actions uniformly at random.

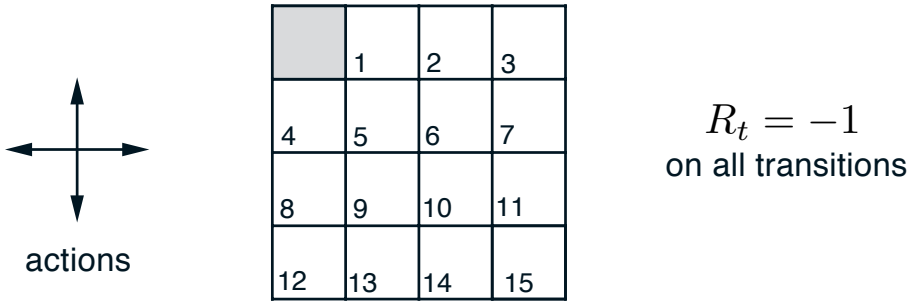


Figure 6: Illustration for grid-world experiment. This figure was generated from figure on page 76 of Sutton and Barto (2018).

To facilitate the analysis, for each state s, we choose the feature vector as

$$\phi(s) = e_s,$$

where $e_s = [0 \cdots 0\ 1\ 0 \cdots 0]$ is the unit vector where the sth element is 1, and all the other elements are 0. In this case, we have

$$V_\theta(s) = \theta^T \phi(s) = \theta.$$

Moreover, as the state space is relatively small, it is convenient for us to get the transition matrix P, and further get

$$\theta^*_{lc} = V^* = (I - \gamma P)^{-1} E[r].$$

Thus we can clearly observe the estimation error $||\bar{\theta}(t) - \theta^*_{lc}||_2$ for each step. To see the impact of the number of agents, we let $N = 100$, 10, 1 in each experiment.

3.1 Constant Step-size with i.i.d. Observation

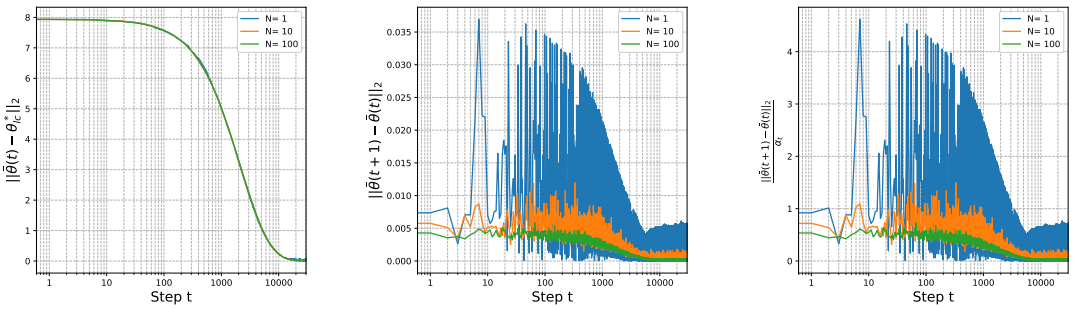
First, we experimented with constant step-size, the experimental results are shown in Figure 7, Figure 8, and Figure 9. It can be observed that $\bar{\theta}(t)$ will converge to θ^*_{lc} in all the scenarios. When we consider the metric $||\bar{\theta}(t+1) - \bar{\theta}(t)||_2$ and $\frac{||\bar{\theta}(t+1) - \bar{\theta}(t)||_2}{\alpha_t}$, we can see the distributed TD(0) algorithm indeed facilitate the convergence process, the larger the value of N is, the faster the convergence would be.

3.2 Decaying Step-size with i.i.d Observation

Next, we experimented with decaying step-size, we consider two different cases, $\alpha_t = \frac{1}{\sqrt{t}}$ and $\alpha_t = \frac{1}{t}$. Figure 10, Figure 11, and Figure 12 show the the experimental results with $\alpha_t = \frac{1}{\sqrt{t}}$. Figure 13 shows the experimental results with $\alpha_t = \frac{1}{t}$. Similar to the results of constant cases, we can see $\bar{\theta}(t)$ will converge to θ^*_{lc} in all the scenarios. Also, when we consider the metric $||\bar{\theta}(t+1) - \bar{\theta}(t)||_2$ and $\frac{||\bar{\theta}(t+1) - \bar{\theta}(t)||_2}{\alpha_t}$, we can see the distributed TD(0) algorithm indeed facilitate the convergence process.

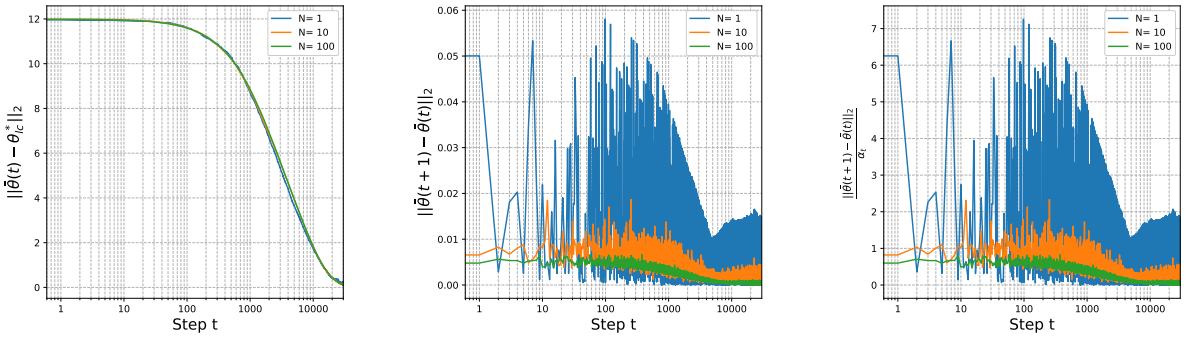
3.3 Results with MDP Observations

Though we have assumed that all the states are sampled i.i.d from a fixed distribution. We also experimented with cases where the states follow a Markov chain observation model. The experimental results are shown in Figure 14 and Figure 15. It can be observed that the results are quite similar to the results with i.i.d assumptions.



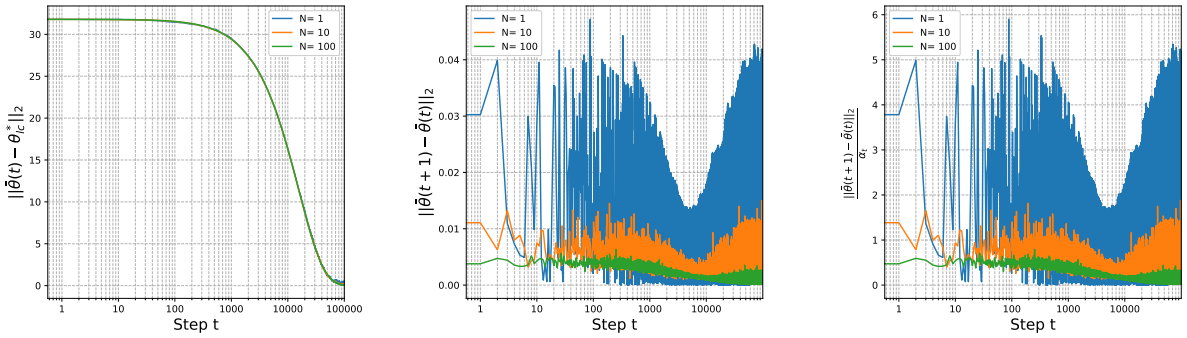
(a) (b) (c)

Figure 7: Experimental results of grid world. **a**, the value of error $||\bar{\theta}(t) - \theta^*_{lc}||_2$ for 30000 steps. **b**, the value of metric $||\bar{\theta}(t+1) - \bar{\theta}(t)||_2$ for 30000 steps. **c**, the value of metric $\frac{||\bar{\theta}(t+1) - \bar{\theta}(t)||_2}{\alpha_t}$ for 30000 steps. The discount factor γ is chosen as 0.5. The step-size is set as $\alpha_t = 0.008$. All the states $S(t)$ are sampled following the i.i.d assumptions.



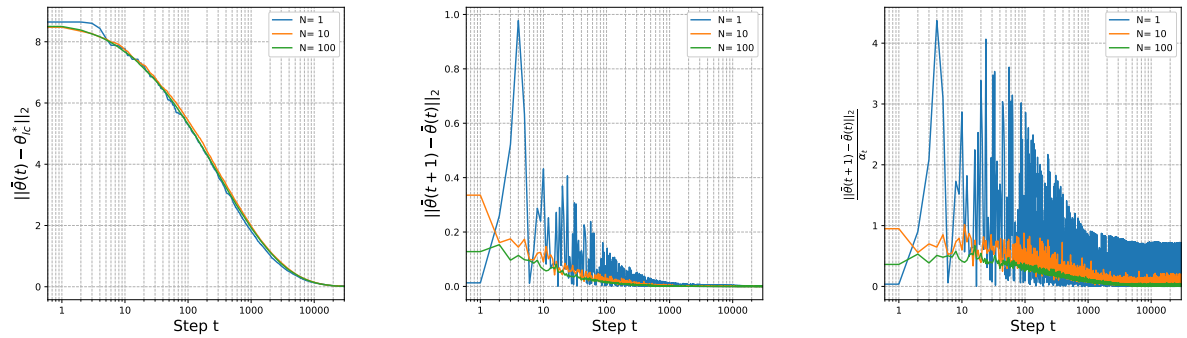
(a) (b) (c)

Figure 8: Experimental results of grid world. **a**, the value of error $||\bar{\theta}(t) - \theta^*_{lc}||_2$ for 30000 steps. **b**, the value of metric $||\bar{\theta}(t+1) - \bar{\theta}(t)||_2$ for 30000 steps. **c**, the value of metric $\frac{||\bar{\theta}(t+1) - \bar{\theta}(t)||_2}{\alpha_t}$ for 30000 steps. The discount factor γ is chosen as 0.75. The step-size is set as $\alpha_t = 0.008$. All the states $S(t)$ are sampled following the i.i.d assumptions.



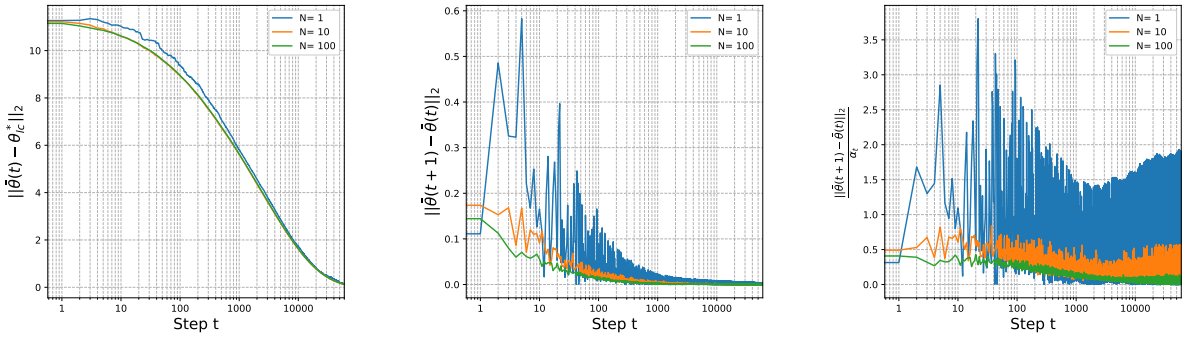
(a) (b) (c)

Figure 9: Experimental results of grid world. **a**, the value of error $||\bar{\theta}(t) - \theta^*_{lc}||_2$ for 100000 steps. **b**, the value of metric $||\bar{\theta}(t+1) - \bar{\theta}(t)||_2$ for 100000 steps. **c**, the value of metric $\frac{||\bar{\theta}(t+1) - \bar{\theta}(t)||_2}{\alpha_t}$ for 100000 steps. The discount factor γ is chosen as 0.9. The step-size is set as $\alpha_t = 0.008$.All the states $S(t)$ are sampled following the i.i.d assumptions.



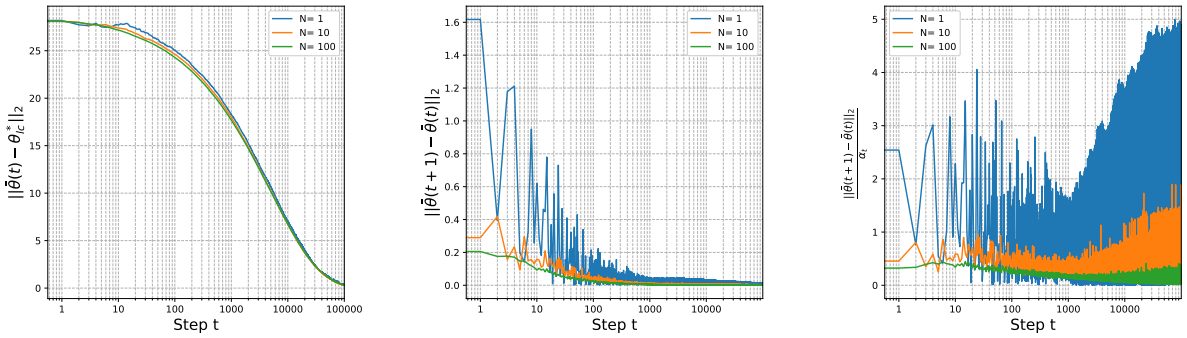
(a) (b) (c)

Figure 10: Experimental results of grid world. **a**, the value of error $||\bar{\theta}(t) - \theta^*_{lc}||_2$ for 30000 steps. **b**, the value of metric $||\bar{\theta}(t+1) - \bar{\theta}(t)||_2$ for 30000 steps. **c**, the value of metric $\frac{||\bar{\theta}(t+1) - \bar{\theta}(t)||_2}{\alpha_t}$ for 30000 steps. The discount factor γ is chosen as 0.5. The step-size is set as $\alpha_t = \frac{0.5}{\sqrt{t+1}}$. All the states $S(t)$ are sampled following the i.i.d assumptions.



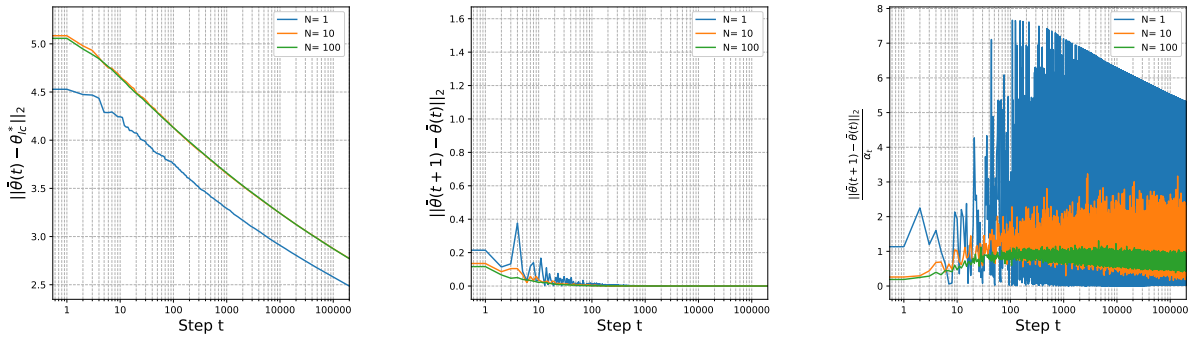
(a) (b) (c)

Figure 11: Experimental results of grid world. **a**, the value of error $||\bar{\theta}(t) - \theta^*_{lc}||_2$ for 60000 steps. **b**, the value of metric $||\bar{\theta}(t+1) - \bar{\theta}(t)||_2$ for 60000 steps. **c**, the value of metric $\frac{||\bar{\theta}(t+1) - \bar{\theta}(t)||_2}{\alpha_t}$ for 60000 steps. The discount factor γ is chosen as 0.75. The step-size is set as $\alpha_t = \frac{0.5}{\sqrt{t+1}}$. All the states $S(t)$ are sampled following the i.i.d assumptions.



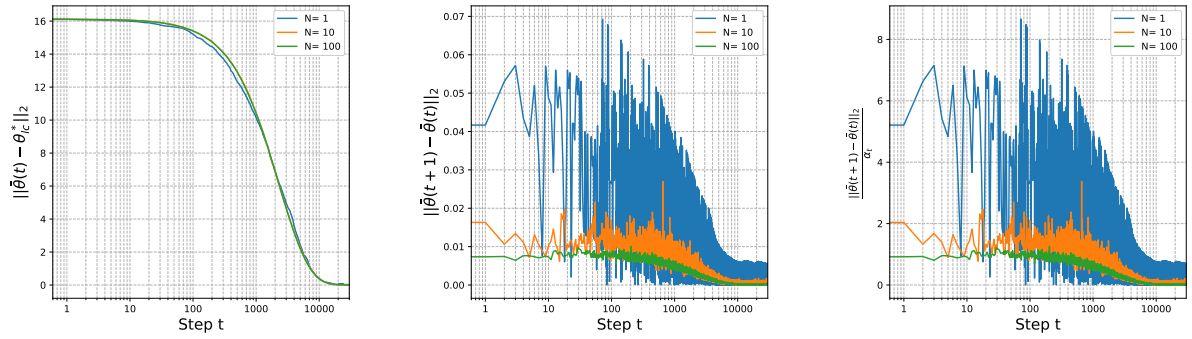
(a) (b) (c)

Figure 12: Experimental results of grid world. **a**, the value of error $||\bar{\theta}(t) - \theta^*_{lc}||_2$ for 100000 steps. **b**, the value of metric $||\bar{\theta}(t+1) - \bar{\theta}(t)||_2$ for 100000 steps. **c**, the value of metric $\frac{||\bar{\theta}(t+1) - \bar{\theta}(t)||_2}{\alpha_t}$ for 100000 steps. The discount factor γ is chosen as 0.9. The step-size is set as $\alpha_t = \frac{0.9}{\sqrt{t+1}}$. All the states $S(t)$ are sampled following the i.i.d assumptions.



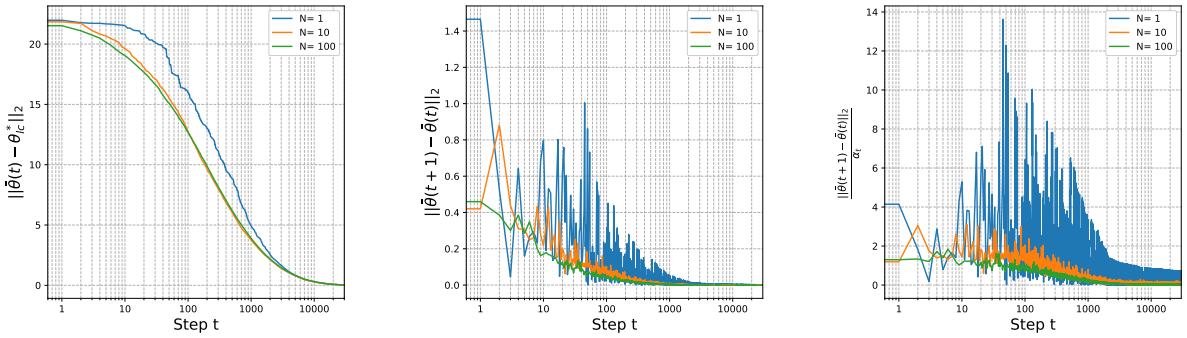
(a) (b) (c)

Figure 13: Experimental results of grid world. **a**, the value of error $||\bar{\theta}(t) - \theta^*_{lc}||_2$ for 200000 steps. **b**, the value of metric $||\bar{\theta}(t+1) - \bar{\theta}(t)||_2$ for 200000 steps. **c**, the value of metric $\frac{||\bar{\theta}(t+1) - \bar{\theta}(t)||_2}{\alpha_t}$ for 200000 steps. The discount factor γ is chosen as 0.5. The step-size is set as $\alpha_t = \frac{0.9}{t+1}$. All the states $S(t)$ are sampled following the i.i.d assumptions.



(a) (b) (c)

Figure 14: Experimental results of grid world. **a**, the value of error $||\bar{\theta}(t) - \theta^*_{lc}||_2$ for 30000 steps. **b**, the value of metric $||\bar{\theta}(t+1) - \bar{\theta}(t)||_2$ for 30000 steps. **c**, the value of metric $\frac{||\bar{\theta}(t+1) - \bar{\theta}(t)||_2}{\alpha_t}$ for 30000 steps. The discount factor γ is chosen as 0.75. The step-size is set as $\alpha_t = 0.008$. All the states $S(t)$ are sampled following the prescribed Markov chain observation model.



(a) (b) (c)

Figure 15: Experimental results of grid world. **a**, the value of error $||\bar{\theta}(t) - \theta^*_{lc}||_2$ for 30000 steps. **b**, the value of metric $||\bar{\theta}(t+1) - \bar{\theta}(t)||_2$ for 30000 steps. **c**, the value of metric $\frac{||\bar{\theta}(t+1) - \bar{\theta}(t)||_2}{\alpha_t}$ for 30000 steps. The discount factor γ is chosen as 0.75. The step-size is set as $\alpha_t = \frac{0.5}{\sqrt{t+1}}$. All the states $S(t)$ are sampled following the prescribed Markov chain observation model.

4 Conclusion

In our experiments, the distributed TD(0) with local state do converge faster when it has more agents in the system, although it is not a linear time speedup. From the experiments on the classic control problems in the OpenAI Gym, we can observe that, one can get more benefit from parallelism with constant step-size than that with decaying step-size. Also, benefit from parallelism is diminishing when we increase agent number. For the experiments on the grid world MDP, we can get θ^*_{lc} exactly. However, when we compare the value of $||\bar{\theta}(t) - \theta^*_{lc}||_2^2$ for different number of agents, their performance are almost the same.

References

Liu, R. and Olshevsky, A. (2021). Distributed td (0) with almost no communication. *arXiv preprint arXiv:2104.07855*.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT Press.