Federated Q-Learning with Reference-Advantage Decomposition: Almost Optimal Regret and Logarithmic Communication Cost

Zhong Zheng, Haochen Zhang and Lingzhou Xue

Department of Statistics,
the Pennsylvania State University

April 26, 2025

Federated Q-Learning Setup

- In this work, we consider a federated Q-learning setting with a central server and M agents. Each agent interacts with a tabular episodic Markov Decision Process (MDP) independently in parallel.

Federated Q-Learning Setup

- In this work, we consider a federated Q-learning setting with a central server and M agents. Each agent interacts with a tabular episodic Markov Decision Process (MDP) independently in parallel.
- **Tabular Episodic Markov Decision Process (MDP)**

  In a tabular episodic MDP $(\mathcal{S}, \mathcal{A}, H, \mathbb{P}, r)$:
  - $\mathcal{S}$: state space, $\mathcal{A}$: action space, H: number of steps.
  - $\mathbb{P} := \{\mathbb{P}_h\}_{h=1}^{H}$ is the time-inhomogeneous transition kernel.
  - $r := \{r_h\}_{h=1}^{H}$ is the collection of reward functions.

Policy and Value Functions

- A policy π is a collection of H functions $\left\{\pi_h : \mathcal{S} \to \Delta^{\mathcal{A}}\right\}_{h\in[H]}$, where $\Delta^{\mathcal{A}}$ is the set of probability distributions over $\mathcal{A}$.

Policy and Value Functions

- A policy π is a collection of H functions $\left\{\pi_h : \mathcal{S} \to \Delta^{\mathcal{A}}\right\}_{h\in[H]}$, where $\Delta^{\mathcal{A}}$ is the set of probability distributions over $\mathcal{A}$.
- We use $Q_h^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and $V_h^\pi : \mathcal{S} \to \mathbb{R}$ to denote the state-action value function and the state value function at step h under policy π.

$$Q_h^\pi(s, a) := r_h(s, a) + \sum_{h'=h+1}^{H} \mathbb{E}_{(s_{h'}, a_{h'})\sim(\mathbb{P},\pi)} \left[r_{h'}(s_{h'}, a_{h'}) \mid s_h = s, a_h = a\right].$$

$$V_h^\pi(s) := \sum_{h'=h}^{H} \mathbb{E}_{(s_{h'}, a_{h'})\sim(\mathbb{P},\pi)} \left[r_{h'}(s_{h'}, a_{h'}) \mid s_h = s\right].$$

Policy and Value Functions

- A policy π is a collection of H functions $\left\{\pi_h : \mathcal{S} \to \Delta^{\mathcal{A}}\right\}_{h\in[H]}$, where $\Delta^{\mathcal{A}}$ is the set of probability distributions over $\mathcal{A}$.
- We use $Q_h^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and $V_h^\pi : \mathcal{S} \to \mathbb{R}$ to denote the state-action value function and the state value function at step h under policy π.

$$Q_h^\pi(s,a) := r_h(s,a) + \sum_{h'=h+1}^{H} \mathbb{E}_{(s_{h'},a_{h'})\sim(\mathbb{P},\pi)}\left[r_{h'}(s_{h'},a_{h'}) \mid s_h = s, a_h = a\right].$$

$$V_h^\pi(s) := \sum_{h'=h}^{H} \mathbb{E}_{(s_{h'},a_{h'})\sim(\mathbb{P},\pi)}\left[r_{h'}(s_{h'},a_{h'}) \mid s_h = s\right].$$

- There always exists an optimal policy $\pi^\star$ for all states and steps. In detail, it achieves the optimal value $V_h^\star(s) = V_h^{\pi^*}(s) = \sup_\pi V_h^\pi(s)$ for all $s \in \mathcal{S}$ and $h \in [H]$.

FedQ-Advantage Algorithm

In this paper, we propose a new algorithm, FedQ-Advantage algorithm. Compared to other federated Q-learning algorithm, it incorporates the following features:

FedQ-Advantage Algorithm

In this paper, we propose a new algorithm, FedQ-Advantage algorithm. Compared to other federated Q-learning algorithm, it incorporates the following features:

1. **Stage-wise update**

   For each state-action-step triple (s, a, h), we divide rounds into different stages. Each stage contains multiple consecutive rounds. The Q-value function for (s, a, h) is updated only at the end of each stage, rather than after every communication round.

FedQ-Advantage Algorithm

In this paper, we propose a new algorithm, FedQ-Advantage algorithm. Compared to other federated Q-learning algorithm, it incorporates the following features:

1. **Stage-wise update**

   For each state-action-step triple (s, a, h), we divide rounds into different stages. Each stage contains multiple consecutive rounds. The Q-value function for (s, a, h) is updated only at the end of each stage, rather than after every communication round.

2. **Upper confidence bound and reference-advantage decomposition**

   We combines two different techniques in single agent RL, UCB and reference-advantage decomposition, to update the Q-value function.

FedQ-Advantage Algorithm

In this paper, we propose a new algorithm, FedQ-Advantage algorithm. Compared to other federated Q-learning algorithm, it incorporates the following features:

1. **Stage-wise update**

   For each state-action-step triple (s, a, h), we divide rounds into different stages. Each stage contains multiple consecutive rounds. The Q-value function for (s, a, h) is updated only at the end of each stage, rather than after every communication round.

2. **Upper confidence bound and reference-advantage decomposition**

   We combines two different techniques in single agent RL, UCB and reference-advantage decomposition, to update the Q-value function.

3. **An optional forced synchronization mechanism**

   Under this mechanism, when one agent triggers the communication condition, the central server terminates the exploration for all agents and begins a new round. Without forced synchronization, the central server waits for each agent to individually meet the communication condition.

Main Results

Define

$$\text{Regret} = \sum_{\text{all episodes } e} \left(V_1^\star(s_{1,e}) - V_1^{\pi_e}(s_{1,e}) \right),$$

where $s_{1,e}$ is the initial state for the episode e.

Main Results

Define

$$\text{Regret} = \sum_{\text{all episodes } e} \left(V_1^{\star}(s_{1,e}) - V_1^{\pi_e}(s_{1,e}) \right),$$

where $s_{1,e}$ is the initial state for the episode e.

Theorem (Regret of FedQ-Advantage)

For FedQ-Advantage algorithm and any $p \in (0,1)$, with probability at least $1-p$, we have

$$\text{Regret}(T) \leq \tilde{O}\left(\sqrt{MSAH^2T} + M\text{poly}(HSA)\right).$$

Here, T is the total number of steps for each agent. $\tilde{O}$ hides logarithmic multipliers on $T, M, H, S, A, 1/p$ and poly *represents some polynomial.*

Main Results

Define

$$\text{Regret} = \sum_{\text{all episodes } e} \left(V_1^\star(s_{1,e}) - V_1^{\pi_e}(s_{1,e}) \right),$$

where $s_{1,e}$ is the initial state for the episode e.

Theorem (Regret of FedQ-Advantage)

For FedQ-Advantage algorithm and any $p \in (0,1)$, with probability at least $1-p$, we have

$$\text{Regret}(T) \leq \tilde{O}\left(\sqrt{MSAH^2T} + M\text{poly}(HSA)\right).$$

Here, T is the total number of steps for each agent. $\tilde{O}$ hides logarithmic multipliers on $T, M, H, S, A, 1/p$ and poly *represents some polynomial.*

This result matches the information lower bound $O(\sqrt{MSAH^2T})$ up to some logarithmic factors.

Main Results

We define the communication cost of an algorithm as the number of scalars (integers or real numbers) communicated between the server and agents.

Main Results

We define the communication cost of an algorithm as the number of scalars (integers or real numbers) communicated between the server and agents.

Theorem (Communication rounds of FedQ-Advantage)

For FedQ-Advantage algorithm with forced synchronization mechanism, the number of communication rounds K satisfies that

$$K \leq MSAH^2 + 4MSAH^2(\log(H) + 3)\log\left(\frac{T}{SAH^3} + 1\right).$$

Without forced synchronization mechanism, the number of communication rounds K satisfies

$$K \leq SAH^2 + 4SAH^2(\log(H) + 3)\log\left(\frac{T}{SAH^3} + 1\right).$$

Main Results

We define the communication cost of an algorithm as the number of scalars (integers or real numbers) communicated between the server and agents.

Theorem (Communication rounds of FedQ-Advantage)

For FedQ-Advantage algorithm with forced synchronization mechanism, the number of communication rounds K satisfies that

$$K \leq MSAH^2 + 4MSAH^2(\log(H) + 3)\log\left(\frac{T}{SAH^3} + 1\right).$$

Without forced synchronization mechanism, the number of communication rounds K satisfies

$$K \leq SAH^2 + 4SAH^2(\log(H) + 3)\log\left(\frac{T}{SAH^3} + 1\right).$$

Since the total number of communicated scalars is $O(MHS)$ in each round, this theorem implies that FedQ-Advantage algorithm achieves a logarithmic communication cost.

Thank you for listening!

Scan for more details about our paper