Privacy-preserving Q-Learning with Functional Noise in Continuous Spaces

Baoxiang Wang (bxiangwang@gmail.com), Nidhi Hegde (nidhi.hegde@borealisai.com)

https://arxiv.org/abs/1901.10634

BOREALIS AI

Summary

- **Privacy-preserving Q-learning protects the reward function from being distinguished. Methods like inverse reinforcement learning cannot learn the reward function.**
- **We achieve so by iteratively adding functional noise to the value function. Differential privacy guarantee and utility analysis are shown.**

Motivation and Problem Settings

Reinforcement learning: RL has been applied widely to real-world tasks including games, advertisements, and recommendations.

Reward function and personal information: The RL models are usually trained using a simulator, which is built upon historical data, e.g. purchasing history, clickthrough history.

Inference of such information: Methods like inverse RL and membership inference can be used to infer the underlying reward function from the released model. This is threatening to the users' privacy.

Continuous state spaces: The states are assumed to be continuous in general, which is aligned with real-world scenarios. A neural network is desired for value function approximation.

Indistinguishability of the reward function: Our objective is to preserve the users' privacy by protecting the reward function. Namely, by observing the output Q-function the attacker cannot distinguish which reward function of $r(\cdot)$ and $r'(\cdot)$ is used, as long as $\|r - r'\|_\infty \leq 1$.

High-privacy regime: We target $\epsilon \leq 0.9$ and $\delta \leq 10^{-4}$ for (ϵ, δ)-differential privacy. With this target, previous methods will not work even on simple examples.

Intuition behind the Algorithm

- **Major difficulty**: The reward signal $r(s, a)$ can appear at any s, and all the reward signals can be different under r and r'.
- **Consequence**: We will need a stronger mechanism of privacy that does not rely on the supervised learning setting where at most one data point in a finite dataset is different.
- **Our approach**: Naturally, we treat a function as one "data point", adding a Gaussian process noise to the Q-function each time it is updated.

Differential Privacy for Vectors and Functions

Definition 1. A randomized mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{U}$ satisfies (ϵ, δ)-differential privacy if for any two neighboring inputs d and d' and for any subset of outputs $\mathcal{Z} \subseteq \mathcal{U}$ it holds that $\mathbb{P}(\mathcal{M}(d) \in \mathcal{Z}) \leq \exp(\epsilon)\mathbb{P}(\mathcal{M}(d') \in \mathcal{Z}) + \delta$.

Definition 2. For all pairs $d, d' \in \mathcal{D}$ of neighboring inputs, the sensitivity of a mechanism $\mathcal{M}$ is defined as $\Delta_\mathcal{M} = \sup_{d,d' \in \mathcal{D}} \|\mathcal{M}(d) - \mathcal{M}(d')\|$, where $\|\cdot\|$ is a norm function defined on $\mathcal{U}$.

Gaussian mechanism discusses under $\mathcal{U} = \mathbb{R}^n$ and ℓ^2-norm $\|\cdot\|_2$ in $\Delta_\mathcal{M}$.

Proposition 3 (Theorem A.1 of [DR14]). If $0 < \epsilon < 1$ and $\sigma \geq \sqrt{2\ln(1.25/\delta)}\Delta_\mathcal{M}/\epsilon$, then $\mathcal{M}(d)+y$ is (ϵ,δ)-differentially private, where y is drawn from $\mathcal{N}(0, \sigma^2 I)$.

Gaussian process mechanism is a functional mechanism, where $\mathcal{U}$ is an RKHS and $\|\cdot\|$ in $\Delta_\mathcal{M}$ is the RKHS norm $\|\cdot\|_\mathcal{H}$.

Proposition 4 (Proposition 8 of [HRW13]). If $0 < \epsilon < 1$ and $\sigma \geq \sqrt{2\ln(1.25/\delta)}\Delta_\mathcal{M}/\epsilon$, then $\mathcal{M}(d)+g$ is (ϵ,δ)-differentially private, where g is drawn from $\mathcal{G}(0, \sigma^2 K)$ and $\mathcal{U}$ is an RKHS with kernel function K.

Algorithm 1 Differentially Private Q-Learning with Functional Noise

Input: the reward function $r(\cdot)$
Parameters: target privacy (ϵ, δ), $\sigma = \sqrt{2(T/B)\ln(e + \epsilon/\delta)}C(\alpha, k, L, B)/\epsilon$
Initialization: linked list $\hat{g}_k[B][2] = \{\}$ of $(s, g(s))$-tuples
for j, b **in** $[T/B] \times [B]$ **do**
  $t \leftarrow jT/B + b$;
  Execute $a_t = \arg\max_a Q_\theta(s_t, a) + \hat{g}_a(s_t)$;
  Receive r_t and s_{t+1}, $s \leftarrow s_{t+1}$;
  $\hat{g}_k[B][2] = \{\}$ when $j \equiv 0 \mod T/JB$;
  Insert s to $\hat{g}_a[:][1]$ such that the list remains monotonically increasing;
  Sample $z_{at} \sim \mathcal{N}(\mu_{at}, \sigma d_{at}))$;
  Update the list $\hat{g}_a(s) \leftarrow z_{at}$;
  $y_t \leftarrow r_t + \gamma \max_a Q_\theta(s_{t+1}, a) + \hat{g}_a(s_{t+1})$;
  $l_t \leftarrow \frac{1}{2}(Q_\theta(s_t, a_t) + \hat{g}_a(s_t) - y_t)^2$;
  For each batch j run one step SGD $\theta \leftarrow \theta + \alpha \frac{1}{B} \nabla_\theta \sum_{t=jB}^{(j+1)B} l_t$;
end for
Output: action-state value function $Q_\theta(s, a)$

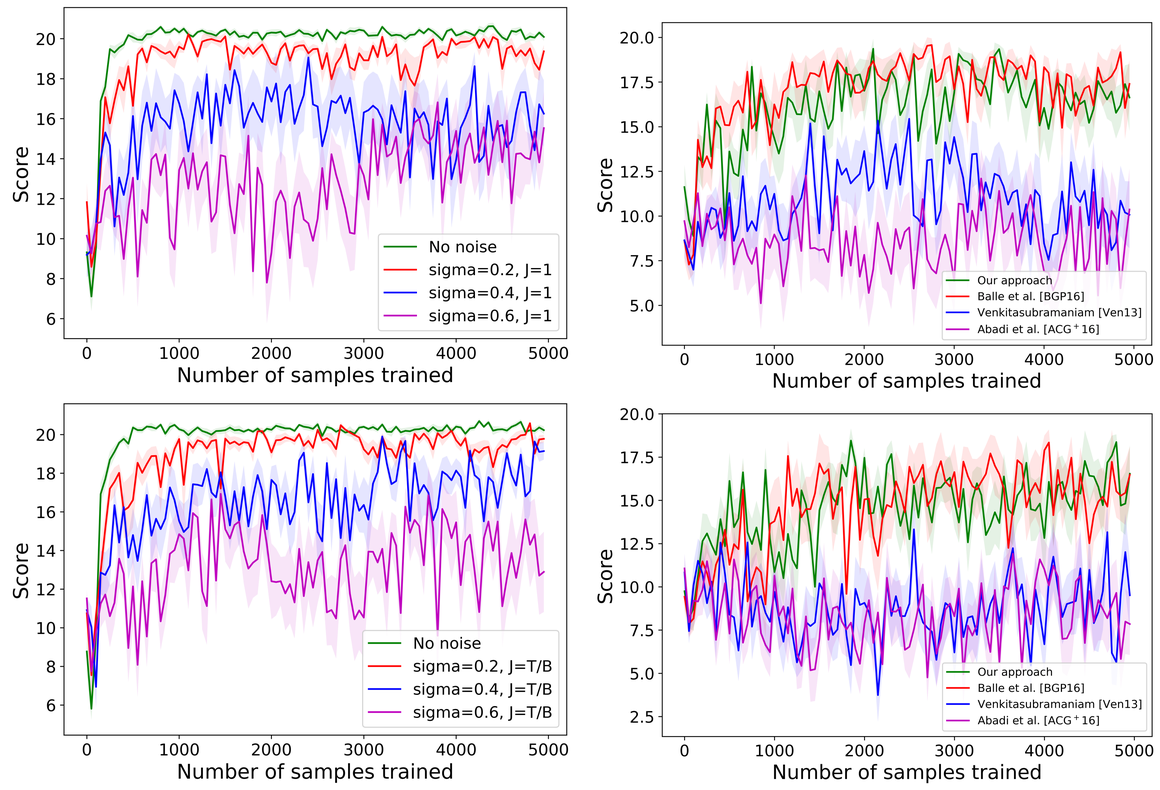
Theoretical Results

Theorem 5. The Q-learning algorithm above is $(\epsilon, \delta + J\exp(-(2k - 8.68\sqrt{\beta}\sigma)^2/2))$-DP with respect to two neighboring reward functions $\|r - r'\|_\infty \leq 1$, provided that $2k > 8.68\sqrt{\beta}\sigma$, and $\sigma \geq \sqrt{2(T/B)\ln(e + \epsilon/\delta)}C(\alpha, k, L, B)/\epsilon$.

Proposition 9. The noised Q-function, during either training or released, responds to N queries in $\mathcal{O}(N\ln(N))$ time.

Proposition 10. Let v' and v^* be the value function learned by our algorithm and the optimal value function, respectively. In the case $J = 1$, $|S| = n < \infty$, and $\gamma < 1$, the utility loss of the algorithm satisfies $\mathbb{E}[\frac{1}{n}\|v' - v^*\|_1] \leq \frac{2\sqrt{2}\sigma}{\sqrt{n\pi}(1-\gamma)}$.

Empirical Results



(a) Performance with different noise levels. Top: $J = 1$; bottom: $J = T/B$.

(b) Comparisons with other approaches. Top: $\epsilon = 0.9$, $\delta = 10^{-4}$; bottom: $\epsilon = 0.45$, $\delta = 10^{-4}$.

References

[DR14] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[HRW13] Rob Hall, Alessandro Rinaldo, and Larry Wasserman. Differential privacy for functions and functional data. *Journal of Machine Learning Research*, 14(Feb):703–727, 2013.