



Article

Differentially Private Actor and Its Eligibility Trace

Kanghyeon Seo and Jihoon Yang *

Machine Learning Research Laboratory, Department of Computer Science and Engineering, Sogang University, Seoul 04107, Korea; sdsdsf123@sogang.ac.kr

* Correspondence: yangjh@sogang.ac.kr

Received: 14 August 2020; Accepted: 8 September 2020; Published: 10 September 2020



Abstract: We present a differentially private actor and its eligibility trace in an actor-critic approach, wherein an actor takes actions directly interacting with an environment; however, the critic estimates only the state values that are obtained through bootstrapping. In other words, the actor reflects the more detailed information about the sequence of taken actions on its parameter than the critic. Moreover, their corresponding eligibility traces have the same properties. Therefore, it is necessary to preserve the privacy of an actor and its eligibility trace while training on private or sensitive data. In this paper, we confirm the applicability of differential privacy methods to the actors updated using the policy gradient algorithm and discuss the advantages of such an approach with regard to differentially private critic learning. In addition, we measured the cosine similarity between the differentially private applied eligibility trace and the non-differentially private eligibility trace to analyze whether their anonymity is appropriately protected in the differentially private actor or the critic. We conducted the experiments considering two synthetic examples imitating real-world problems in medical and autonomous navigation domains, and the results confirmed the feasibility of the proposed method.

Keywords: differential privacy; reinforcement learning; actor-critic; eligibility trace

1. Introduction

Reinforcement learning (RL) defines the steps and procedures required to map situations to actions aiming to maximize a accumulated reward signal [1] and serves as a practical framework for decision-making problems. RL can be applied to a variety of data types. In particular, it can be used for sensitive or private data, such as a patient's treatment procedure or trajectories for car navigation. However, with the development and deployment of diverse RL-based technologies in computer science, the demand for these private or sensitive data increases. Rather than using these raw data as it is, it is needed to prevent personal privacy leakage while maintaining the original data's utility. Difference privacy (DP), first proposed by Dwork [2], is a privacy model that is used to design a mechanism to prevent the leakage of personal information due to multiple queries in a database storing sensitive data. Based on the DP concepts, many studies aimed at applying deep learning have been conducted. In particular, Abadi et al. [3] have proposed incorporating the calibrated noise into the stochastic gradient descent computation to protect the privacy of training data.

Many prior RL works focused on DP have also been reported. First, Balle et al. [4] presented a differentially private algorithm for an on-policy evaluation concerning synthetic medical domain problems. Lebensold et al. [5] recently extended their work to actor-critic algorithm [6,7] with differentially private critics. They realized output perturbation [8] to guarantee privacy while adding randomized noise to all elements of an estimated state value vector in the least-squares policy evaluation method. Xie et al. [9] also introduced an application of off-policy evaluation to a differential privacy algorithm. They added noise to a stochastic gradient descent update [10,11] aiming

to guarantee privacy using the off-policy evaluation in the GTD2 [12] algorithm. Wang and Hegde [13] suggested a differentially private Q-learning algorithm in a continuous space to preserve a privacy of the value function approximator by adding a Gaussian process noise to the value function at each iteration during training.

However, most of the previous studies focused mainly on value-based algorithms and did not consider policy gradient and eligibility traces. Therefore, in the present research, we determine whether DP can be applied to the actor-critic algorithm, a representative policy-based method. In this approach, an actor takes actions and updates its parameters by directly interacting with an environment; however, the critic can estimate the state value only through bootstrapping. Although the state value function is necessary to learn the parameters of an actor *u* it is not used to perform action selection. Instead, the actor learns the parameters in the direction of increasing the performance function J(u)computed when the actor follows the policy [1]. Therefore, the actor that stores and updates its actions requires more information compared with the critic that only performs policy evaluation. In addition, an eligibility trace is a short-term memory vector that records the credit according to the frequency and recency [14] of visited states or actions during each episode. The corresponding eligibility traces of the actor and the critic have the same properties corresponding to recording and updating information from the learning process. Therefore, if RL relies on private and sensitive data, it is necessary to ensure that the parameters and the eligibility trace of the actor are not leaked during training. For example, in medical or automatic navigation domains, the parameters and its eligibility trace updated by an actor may indicate the recovery state of a patient and reveal the prescription defined by a doctor in the medical domain. Alternatively, they may exhibit the information representing the locations of a driver and the corresponding traces of movements in the car navigation domain. Furthermore, their gradient used in the training process may represent a private tendency existing in the information (the recovery rate of a specific patient in the medical domain or the preferred private route of a user in the navigation domain). In this paper, we propose a method to protect the privacy of sensitive data corresponding to an actor and its eligibility trace during training in the actor-critic approach. To realize this, we apply the gradient perturbation method introduced by Abadi et al. [3] to the off-policy actor-critic [15] with some modifications.

The main contributions of this paper can be summarized as follows. We verify that the policy-based algorithm also guarantees DP and, more importantly, protects the privacy of its eligibility trace. In other words, we confirm the applicability of differential privacy to the actors updated using the policy gradient method and demonstrate its superiority compared with differentially private critic learning. Furthermore, we measure cosine similarity between the eligibility trace vectors of the differentially private and non-differentially private approaches to analyze whether their anonymity is protected concerning the differentially private actor or the critic. We also evaluate the experimental results corresponding to two synthetic examples imitating real-world problems in medical and autonomous navigation domains, thereby demonstrating the feasibility of the proposed method. We note explicitly that although in the work reported by Lebensold [5] (AC with DP critic) may deem similar to the proposed approach in terms of using the actor-critic algorithm, it differs in two aspects. First, the aim of their work was to guarantee DP to transfer learning for sample inefficient RL. However, we do not consider transfer learning in our works. Instead, we aim to guarantee the DP of an actor and its eligibility trace in online learning. Second, they implemented a trusted data aggregator (producer) providing a state value function guaranteed by DP-LSL [4] so that the state value was estimated using the first-visit Monte Carlo method without considering an eligibility trace for several untrusted agents (consumer). Then, the consumers used this state value function as a critic in the actor-critic algorithm. Consequently, their work was focused only on guaranteeing the DP of the critic, not an actor. In contrast, we focus on guaranteeing DP in the policy-based algorithm and propose a differentially private actor and its eligibility traces, instead of considering the critic.

The rest of this paper is organized as follows. We introduce the background concepts used in this work and provide the details of the proposed approach in Section 2. In Section 3, we discuss the

details of the conducted experiments, providing results and discussions. Finally, we conclude this work outlining limitations in Section 5.

2. Background

In this section, we introduce the essential definitions of the concepts used in this work: differential privacy and an off-policy actor-critic.

2.1. Differential Privacy

Differential privacy (DP) [2,8,16], introduced by Dwork et al. provides a robust standard framework to guarantee privacy in aggregated databases and prevents from various adversarial attackers. DP relies on the concept of two neighboring data sets that differ only by a single element.

Definition 1. A randomized mechanism M including the domain D and range R; $M:D\to R$ satisfies (ϵ, δ) -differential privacy if, for any two adjacent inputs $d, d'\in D$ and for any subset of outputs $S\subseteq R$, the following holds:

$$Pr[M(d) \in S] \le e^{\epsilon} Pr[M(d') \in S] + \delta$$
 (1)

To approximate deterministic real-valued function $f:D\to R$ with a differential privacy mechanism, we incorporate additive noise calibrated to the sensitivity of f that is defined as the maximum of the absolute distance |f(d)-f(d')|, where d, d' are adjacent input data sets. Selecting between the widely used Gaussian and Laplace noise mechanisms [2,8,16], in the present study, we employ the Gaussian mechanisms defined as follows:

$$M(d) \triangleq f(d) + N(0, \sigma^2) \tag{2}$$

where $N(0,\sigma^2)$ is a normal (Gaussian) distribution with a mean of 0 and a standard deviation σ . To achieve (ϵ, δ) -differential privacy, we set the $\sigma = \sqrt{2log(1.25)/\delta} \frac{s_f}{\epsilon}$ [2]. We consider the deterministic real value function $f: D \to R$ as a gradient of performance of an actor $(\nabla_u log \pi_u(a|s) = \nabla_u \pi_u(a|s)/\pi_u(a|s))$ which is a probability of taking an action actually taken divided by the probability of taking that action [1].

2.2. Off-Policy Actor-Critic

Off-policy actor-critic, introduced by Degris et al. [15], integrates properties of off-policy learning into the adaptability of action selection in the actor-critic framework (Algorithm 1).

Algorithm 1 Off-Policy Actor-Critic

```
1: Initialize the vectors e_u, e_v and w to zero
                                                                                        //eligibility traces for actor (e_u) and critic (e_v)
     Initialize the vectors u and v arbitrarily
                                                                                        //parameters for actor (u) and critic (w, v)
      Initialize a state s
     for each step do
 4:
            Choose an action, a, according to b(.|s)
           Observe resultant reward, r', and next state, s'
 6:
           \delta \leftarrow r + \gamma(s')v^t x_{s'} - v^t x_s\rho \leftarrow \pi_u(a|s)/b(a|s)
 7:
 8:
           Update the critic (GTD(\lambda) algorithm):
10:
                   \begin{array}{l} e_v \leftarrow \rho(x_s + \gamma(s)\lambda e_v), \\ v \leftarrow v + \alpha_v [\delta e_v - \gamma(s)(1 - \lambda)(w^t e_v)x_s] \\ w \leftarrow w + \alpha_w [\delta e_v - (w^t x_s)x_s] \end{array}
12:
13:
15:
           Update the actor:
                   e_u \leftarrow \rho[\nabla_u log \pi_u(a|s) + \gamma(s) \lambda e_u]
16:
                   u \leftarrow u + \alpha_u \delta e_u
18: end for
```

The actor learns the policy parameter u according to the temporal difference error estimated by the critic (GTD(λ) [17]). In this study, the linear function approximation has been used to estimate the state values of $V^{\pi_r}(s): \hat{V}(s) = v^t x_s$, where $x_s \in \mathbb{R}^{N_v}$, $N_v \in \mathbb{N}$ is the feature vector of state s and $v \in \mathbb{R}^{N_v}$ is the other weight vector [15]. Using the importance sampling method, the actor and critic are made convenient to update their parameters even if the behavior policy (b(a|s)) that is not the target policy $(\pi_u(a|s))$ selects the action.

In this algorithm, there are two kinds of eligibility traces for actor e_u and critic e_v . In the case of the critic, it serves as an auxiliary to update a parameter of the state value function w. In the case of the actor, it is used to update the parameter u of a policy. Specifically, the critic acts only as a baseline for the actions taken by an actor so that the eligibility trace of an actor has a more direct effect in selecting actions than that of the critic.

Therefore, in this study, we incorporate calibrated noise into the actor's gradient to protect the privacy of an eligibility trace. As we aim to determine how incorporating noise into an actor directly affects the target policy and its eligibility trace, we employ the ϵ -greedy policy [1] instead of the policy used in the off-policy actor-critic approach. More details are provided in Section 4.

3. Differentially Private Actor and Its Eligibility Trace Learning

In this section, we provide the details of the proposed algorithm: differentially private actor and its eligibility trace (DP-Actor) illustrated in Algorithm 2.

Algorithm 2 Differentially Private Actor and Its Eligibility Trace Learning

```
1: Initialize a weight vector w to zero
 2: Initialize weight vectors v, u arbitrarily
    for each episode do
          Initialize eligibility trace vectors e_v and e_u to zero
 4:
          Initialize a state a
 5:
 6:
 7:
          for each step do
               Choose an action, a by \epsilon – Greedy
 8:
               if a is selected by b(.|s) : \rho \leftarrow \pi_u(a|s)/b(a|s)
 9:
10:
               else : \rho \leftarrow 1
11:
               Observe resultant reward, r, and next state, s'
12:
               \delta \leftarrow r + \gamma(s')v^t x_{s'} - v^t x_s
13:
14:
               Update the critic (GTD(\lambda) algorithm):
15:
                      e_v \leftarrow \rho(x_s + \gamma(s)\lambda e_v)
16:
                      v \leftarrow v + \alpha_v [\delta e_v - \gamma(s')(1 - \lambda)(w^t e_v)x_s]
17:
                      w \leftarrow w + \alpha_w [\delta e_v - (w^t x_s) x_s]
18:
19:
               Update the noised actor:
20:
                      g(x) \leftarrow \nabla_u log \pi_u(a|s) / (\|\nabla_u log \pi_u(a|s)\|_2 / |C|)
21:
22:
                      \widetilde{g(x)} \leftarrow \overline{g(x)} + N(0, \sigma^2 C^2 I)
23:
24:
                      e_u \leftarrow \rho[g(x) + \gamma(s)\lambda e_u]
25:
                      u \leftarrow u + \alpha_u \delta e_u
26:
27:
               s \leftarrow s'
28:
          end for
29:
30: end for
```

Electronics 2020, 9, 1486 5 of 13

Our work is based on prior works [3,15] with several modifications. First, although Abadi et al. in [3] guaranteed DP in a non-convex objective function, we aim to provide DP in a convex objective one, as we use linear function approximation to estimate value functions, according to Degris et al. [15]. The Gaussian distribution is used to generate a calibrated noise vector for the stochastic gradient ascent update of the actor. The loss function that serves as a measurement for the performance of policy (π_u) is defined as J(u)). Its gradient $(\nabla_u log \pi_u(a|s))$ corresponds to the deterministic real-value function $f: D \to R$ in Equation (2). We fix the mini-batch size equal to 1. Incorporating the noise vector generated based on the Gaussian mechanism into actor learning consequently influences its corresponding eligibility trace vector e_u that contains the information about how recently and frequently an agent has been visited according to actions taken by an actor. Second, as the proposed approach is based on linear function approximation used to estimate the value function $(\hat{V}(s) = v^t x_s)$, where all parameters are represented as vectors, and each element of a gradient includes noised corresponding to different values, we normalize the gradient vector and amplify it using |C| factor that is defined as a median of a gradient in each update process. Third, we modify the policy defining it as the ϵ -greedy instead of Degris et al. [15]. In their work, the behavior policy (b(a|s)) totally selects actions and updates the target policy $(\pi_u(a|s))$ though importance sampling. However, we empirically demonstrate that it is more appropriate to use the ϵ -greedy policy. An action is selected by the target policy $(\pi_u(a|s))$ with a probability of $1 - \epsilon$, whereas the action is selected with the probability of ϵ by the behavior policy (b(a|s)). A uniform random distribution is used in the behavior policy. The importance sampling ratio ρ can be calculated in two ways: as $\pi_u(a|s)/b(a|s)$ if the behavior policy takes action, and as a_t and 1 if the target policy selects action a_t .

The proposed algorithm includes two key steps. First, we obtain a gradient of performance vector $(\nabla_u log \pi_u(a|s))$ of an actor and normalize with its L2-norm and multiply it by the median of absolute gradient value |C|. Then, we add noise into this normalized gradient using a vector sampled from the Gaussian distribution with the following scale:

$$\sigma = \frac{\sqrt{2log(1.25/\delta)}}{\epsilon} \tag{3}$$

Second, the noised gradient of an actor influences the process of updating its eligibility trace e_u . Consequently, the affected eligibility trace accelerates the process of updating policy parameter u that finally impacts the result of selecting actions. Overall, parameter u and its eligibility trace e_u are protected in terms of privacy as a result of adding the noise calibrated in the stochastic gradient ascent during the actor's learning process.

$$u \leftarrow u + \alpha_u \delta e_u$$
 (4)

4. Experiments

In this section, we report the results obtained by applying the proposed algorithm to the two kinds of synthetic toy examples. We conducted a simulation concerning real cases that required protecting privacy. First, as the gradient update occurred in both the actor and the critic, we compared performance in terms of rewards concerning all cases in which the gradient is perturbated by the noise vector. The cases included the proposed model based on a differentially private actor (DP-Actor), a differentially private critic (DP-Critic), and combined differentially private actor and critic (DP-Both). Moreover, we analyzed the non-differentially private (non-DP) model without noise perturbation as a baseline for performance evaluation. Algorithms 3 and 4 illustrate the DP-Critic and DP-Both approaches in which we applied the same method of gradient perturbation as in the DP-Actor. Please note that in these algorithms, $|C_{actor}|$ and $|C_{critic}|$ are absolute medians of gradient at the actor and critic respectively.

Algorithm 3 Differentially Private Critic (DP-Critic)

```
1: Update the noised critic (GTD(\lambda) algorithm with noise):
                e_v \leftarrow \rho(x_s + \gamma(s)\lambda e_v)
 3:
                \overline{z(x)} \leftarrow [\delta e_v - \gamma(s')(1-\lambda)(w^t e_v)x_s]/(||[\delta e_v - \gamma(s')(1-\lambda)(w^t e_v)x_s]||_2/|C_{critic}|)
  4:
  5:
                \widetilde{z(x)} \leftarrow \overline{z(x)} + N(0, \sigma^2 C_{critic}^2 I)
  6:
  7:
                v \leftarrow v + \alpha_v \widetilde{z(x)} 
 w \leftarrow w + \alpha_w [\delta e_v - (w^t x_s) x_s]
 8:
 9.
10:
11: Update the actor:
                e_u \leftarrow \rho[\nabla_u log \pi_u(a|s) + \gamma(s)\lambda e_u] \\ u \leftarrow u + \alpha_u \delta e_u
12:
13:
```

Algorithm 4 Differentially Private Actor and Critic (DP-Both)

```
1: Update the noised critic (GTD(\lambda) algorithm with noise):
                 e_v \leftarrow \rho(x_s + \gamma(s)\lambda e_v)
 3:
                \overline{z(x)} \leftarrow [\delta e_v - \gamma(s')(1-\lambda)(w^t e_v)x_s]/(||[\delta e_v - \gamma(s')(1-\lambda)(w^t e_v)x_s]||_2/|C_{critic}|)
  4:
  5:
                \widetilde{z(x)} \leftarrow \overline{z(x)} + N(0, \sigma^2 C_{critic}^2 I)
  6:
 7:
                 v \leftarrow v + \alpha_v \widetilde{z(x)} \\ w \leftarrow w + \alpha_w [\delta e_v - (w^t x_s) x_s]
  8:
 9:
10:
11: Update the noised actor:
12:
                 \overline{g(x)} \leftarrow \nabla_u log \pi_u(a|s) / (\|\nabla_u log \pi_u(a|s)\|_2 / |C_{actor}|)
13:
14:
                 \widetilde{g(x)} \leftarrow \overline{g(x)} + N(0, \sigma^2 C_{actor}^2 I)
15:
16:
                \begin{array}{l} e_u \leftarrow \rho[\widetilde{g(x)} + \gamma(s)\lambda e_u] \\ u \leftarrow u + \alpha_u \delta e_u \end{array}
18:
```

Second, to ensure that the privacy of eligibility traces was protected appropriately, we measured the cosine similarity of eligibility traces between the non-DP and DP cases (DP-Actor, DP-Critic, and DP-Both). Finally, in the policy improvement part, we explained the reasons for using the ϵ -greedy policy instead of the soft-max one even if our work belongs to policy-gradient algorithms relying on the results of the performance comparison between the two policies.

We set the number of episodes in both environments limited to 500. All parameters were processed according to Table 1 to find the optimal parameter combination with a discount factor $\gamma = 0.99$, $\delta = 10^{-6}$. As we incorporated the calibrated noises into each element of a gradient vector, we set C equal to the median value of the gradient. Therefore, it was proportional to the scale of the gradient of the actor or the critic.

Table 1.	Parameter	Search	Ranges.
----------	-----------	--------	---------

Parameters	Ranges
α_w	{0.001, 0.005, 0.01, 0.05, 0.1, 0.5}
α_{v}	$\{0.001, 0.005, 0.01, 0.05, 0.1, 0.5\}$
α_u	$\{0.001, 0.005, 0.01, 0.05, 0.1, 0.5\}$
λ	{0, 0.2, 0.4, 0.6, 0.8, 0.99}
ϵ	$\{0.1, 0.5, 1, 5, 10\}$

4.1. Experimental Setup

We introduce details of two synthetic examples imitating the real-world problems. One is the environment representing the patient's medical histories (Patient Treatment Progression), and the other (Taxi-v2) imitates a users' trajectories in the automatic navigation environment.

4.1.1. Patient Treatment Progression

According to Lebensold et al. [5], the Markov decision process (MDP) experiments can be considered to be clinical in nature. It means that the patient's treatment and status data is represented as a state vector similar to [18]. We established the setup of MDP similarly as in [5] so that it comprised a chain of 100 states (MDP-100). In each state, an agent can execute two kinds of actions with state transition probabilities: staying or moving to the right. For example, if the agent proceed with the moving-to-the-right action, then, with a probability of 0.1, the agent could stay in its current state or with that of 0.9—advance to the right. The agent received a reward of -1 in each time step and that of 1 in the absorbing state. Although this setup described the policy evaluation in the medical domain [4], it can be considered to be an eligibility trace recording historical states and information about actions, and therefore, it can be used to represent the past progress of a patient in traces of recovery.

4.1.2. Taxi-V2

Figure 1 describes the second practical case corresponded to the navigation tracking domain. We analyzed the case of Taxi-v2 introduced by [19]. Its purpose was to pick up passengers at one of the predefined locations (R, G, Y, or B) and drop them off in a specified destination (R, G, Y, or B). It is a 5 \times 5 grid-world that comprised 500 discrete states corresponding to 25 taxi locations, 5 passenger locations, and 4 destinations. Actions were composed of six discrete sets: {0: move south, 1: move north, 2: move east, 3: move west, 4: pick up a passenger, 5: drop off a passenger}. An agent received -1 reward for each time step, +20 reward for the passenger pickup and drop-off operation executed correctly at proper destinations, and -10 otherwise. This simple environment could be considered to be a database that could store and use trajectories of a person in the automatic navigation domain.

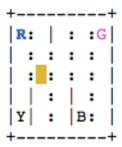


Figure 1. Taxi-v2 in OpenAI gym (https://gym.openai.com/envs/Taxi-v2/).

4.2. Results

We analyze the result of our works from three perspectives: First, we evaluated our works in terms of average reward with various privacy budget values. Second, we measured the anonymity of the eligibility trace vectors when DP was applied by cosine similarity. Finally, we will explain the reasons for using the greedy policy rather than a soft-max one.

4.2.1. Performance and Privacy Budget

Table 2 represents the best parameter combinations in both considered environments (MDP-100 and Taxi-v2). Based on Figure 2, the following three observations were derived. First, the DP-Critic achieved the highest performance in all environments, meaning that the critic exhibited the greatest robustness to noise. Second, the differences in performance estimates corresponding to the

DP-Critic and DP-Actor were larger in Taxi-v2 compared with MDP-100. Third, the learning variances of DP-Both and DP-Actor were greater than those of the DP-Critic (Table 3). The main reasons for these were related to the scale of gradient values and the complexity of the considered environments. We empirically checked that the gradient of the critic ($[e_v - (s^{'})(1-\lambda)(w^te_v)x_s]$) was larger than that of the actor ($\nabla_u log \pi_u(a|s)$), resulting in robustness to perturbation by noise. Accordingly, due to the relatively larger noise impact on the actor's gradient, DP-Actor exhibited a larger learning variance compared with DP-Critic. The complexity of the considered environments also contributed to these phenomena. We assumed that the complexity of an environment relied on the number of states and actions. Accordingly, the case of MDP-100 was deemed less complex (100 states and 2 actions) compared with Taxi-v2 (500 states and 6 actions). Therefore, the behavioral freedom of an agent differed depending on the environment. In other words, with an increase in the number of action types to be selected in a state and a raising number of states, the sensitivity to noise augmented during training because an agent could not select proper actions due to noise in its parameter update process.

Table 2. Optimal Parameters for DP-Actor.

	α_v	α_w	α_u	λ	δ	ϵ
MDP-100	0.01	0.5	0.5	0.8	10^{-6}	10
Taxi-v2	0.05	0.05	0.5	0	10^{-6}	10

Table 3. Performance Standard Deviation.

	Non-DP	DP-Actor	DP-Critic	DP-Both
MDP-100	30.26	68.04	27.1	66.63
Taxi-v2	43.47	57.21	50.03	52.72

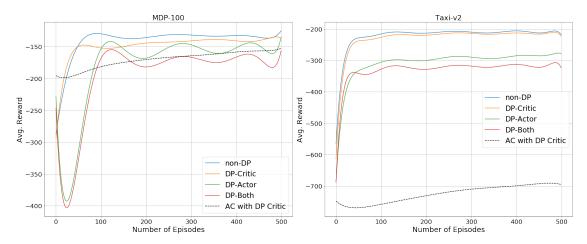


Figure 2. Learning curves in two synthetic examples (MDP-100, Taxi-v2). We choose the optimal parameter combination among all of the combinations after averaging 10 runs for each parameter setting.

We observed the significant difference in performance estimates between the proposed method and AC with DP-Critic in Taxi-v2; however, there are some aspects to consider before comparing these results directly. First, AC with DP-Critic implied using DP-LSL [4], one of the output perturbation methods used to guarantee DP. In the proposed method, instead, we used the gradient perturbation approach. Second, AC with DP-Critic did not employ an eligibility trace that accelerated learning progress, and therefore, it required passing a larger number of episodes to converge to local optimal compared with the proposed method (the authors of [5] set the number of episodes equal to 10,000 in the corresponding experiment).

Electronics 2020, 9, 1486 9 of 13

Figure 3 represents the performance changes corresponding to various privacy budgets (ϵ). Although the performance of DP-Actor represented in Figure 2 was inferior compared with that of DP-Critic, as seen in Figure 3, the performance degradation corresponding to the lower privacy budget in the case of DP-Actor deemed to be more stable compared with DP-Critic. DP-Actor exhibited almost similar performance estimates in a wider privacy budget range ($\epsilon = \{10, 5, 1, 0.5\}$ in MDP-100, and $\epsilon = \{10, 5, 1\}$ in Taxi-v2) compared with the results corresponding to DP-Critic ($\epsilon = \{10, 5, 1\}$ in MDP-100, $\epsilon = \{10, 5\}$ in Taxi-v2). We considered that the reason for this difference was related to a smaller value of |C| for the actor compared with that of the critic. In other words, as we added noise based on $\sigma^2 C^2$, its scale for the actor was relatively smaller compared with that of the critic (due to smaller gradient scale) although the privacy budget decreased in the process of determining the scale of noise. Consequently, DP-Actor had more opportunities for lowering the privacy budget without causing significant performance degradation compared with DP-Critic. We also noted that the performance changes associated with the privacy budget varied depending on the complexity of an environment. The complexity of an environment affected the distribution of performance estimates according to changes in the privacy budget. In MDP-100, their distribution was more cohesive compared with the case of Taxi-v2. Specifically in the DP-Both case with combined DP-Actor DP-Critic, it demonstrated more significant performance degradation from $\epsilon=5$ to $\epsilon=1$ in Taxi-v2 compared with the results corresponding to MDP-100. Therefore, it was required to consider complexity of an environment interacting with an agent to define a proper privacy budget.

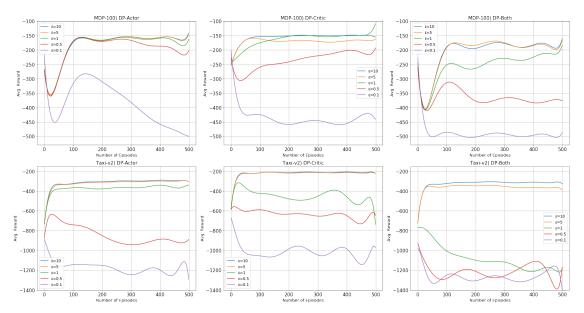


Figure 3. Learning curves according to changes of the privacy budget ($\epsilon = \{10, 5, 1, 0.5, 0.1\}$) in synthetic examples (MDP-100, Taxi-v2). We fixed the optimal parameter set from each case and varied only the privacy budget (ϵ).

4.2.2. Anonymity for the Eligibility Traces

We measured the cosine similarity between the eligibility trace vectors corresponding to non-DP and DP-Actor, DP-Critic, and DP-Both to determine to what extent the privacy of vectors was protected, as shown in Table 4.

Based on the aforementioned observations, we derived the following conclusions. As |C| of the critic was relatively larger than that of the actor, the amount of noise incorporated into the critic was more prominent. Therefore, it was natural to expect that the cosine similarity of critics should have been less than the actor similarity. On the contrary, the cosine similarity of DP-Critic differed considerably from that of DP-Actor with higher similarity values. This was because there was a difference in the update process between the gradients of the actor and the critic, as well as in their

eligibility traces. In the case of the actor, when the gradient perturbed by noise $(g(x) + N(0, \sigma^2C^2I))$ was updated, it directly influenced its eligibility trace $(e_u \leftarrow \rho[g(x) + (s)\lambda e_u])$. However, in the case of the critic, the noised gradient $([e_v - (s')(1 - \lambda)(w^te_v)x_s])$ did not have any impact on its eligibility trace and affected only the bootstrapping parameter v of the critic. The influence of inserted noise was reduced as a result of calculating the TD-error (δ) . Therefore, the cosine similarity in DP-Critic was inevitably higher than that of DP-Actor, meaning that DP-Critic was associated with the greater risk of privacy leakage compared with DP-Actor.

	DP-Actor	DP-Critic	DP-Both
MDP-100	0.021 (0.0042)	0.839 (0.006)	e_u : 0.014 (0.003) e_v : 0.497 (0.006)
Taxi-v2	-0.0008 (0.0003)	0.035 (0.001)	$e_u : -0.0001 (0.0003)$ $e_v : 0.0146 (0.0007)$

Table 4. Mean (Variance) of cosine similarity between non-DP and each cases.

As seen in Figures 2 and 4, and Table 3, the lower the performance due to noise, the lower the similarity of an eligibility trace vector. Therefore, it is necessary to balance between the performance degradation and the degree of anonymity for eligibility traces. We also found that the degree of anonymity for an eligibility trace vector depended on the complexity of an environment according to the fact that with an increase of complexity, the cosine similarity of an overall eligibility trace vector decreased. In particular, DP-Critic exhibited lower similarity in the case Taxi-v2 compared with MDP-100. However, with respect to DP-Actor, there was still a slightly higher risk of privacy leakage for DP-Critic in the case of Taxi-v2. Conversely, although DP-Actor exhibited the performance degradation compared with the performance of the non-DP approach in both the MDP-100 and Taxi-v2 setups, its cosine similarity was maintained at a constant level, indicating that privacy was protected to a certain degree regardless of environments.

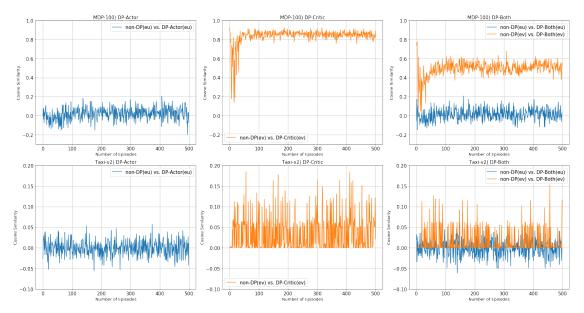


Figure 4. Cosine similarities of eligibility trace vectors between the vectors from non-DP (e_u , e_v) and corresponding vectors from each cases (DP-Actor (e_u), DP-Critic (e_v), and DP-Both (e_u , e_v)). We averaged the vectors generated from same episode over 10 runs to avoid random seed effect.

4.2.3. Reasons of Using the ϵ -Greedy Policy

We parameterize policy with the ϵ -greedy rather than soft-max in action preferences. Although parameterizing policy with the soft-max distribution has more advantages over the ϵ -greedy

policy in policy-gradient algorithm [1], we empirically find that in all DP-applied cases, if we set the policy to be the ϵ -greedy, the DP-Actor only converges to local optimal faster than soft-max one.

Figure 5 illustrates this phenomenon, which can be explained by the fact that the soft-max policy should consider the preference over all other actions with relatively small gradients. This soft-max's property results in the more exploration due to inserted noise compared with the exploration required in the ϵ -greedy policy that considers only one of the largest action preferences with probability (1- ϵ). Therefore, the soft-max policy inevitably convergences to the local optimal slower than the ϵ -greedy one.

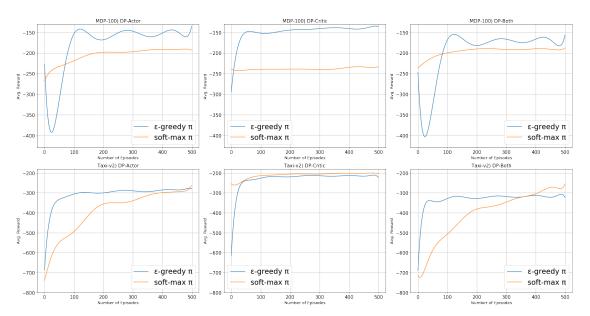


Figure 5. Learning curves by two different policies (ϵ -greedy and soft-max). We select the best performing parameter sets in all of the parameter combinations after averaging rewards for each parameter set run over 10 runs.

5. Conclusions

We proposed a differential-private actor and a learning method for its eligibility trace in the actor-critic approach. Inspired by the Abadi's work [3], we modified off-policy actor-critic algorithm [15], and confirmed that DP was guaranteed in the policy-based algorithm. As a result of experiments, we determined that the DP-Actor was more advantageous compared with the DP-Critic and DP-Both cases. First, although the DP-Actor demonstrated slightly inferior performance compared with that of the DP-Critic, it maintained stable performance within a larger privacy budget range. Second, we confirmed that the anonymity level of eligibility trace vectors was higher in the case of using DP-actor compared with the other cases. We measured cosine similarities for the degree of anonymity between eligibility trace vectors concerning the considered DP cases and one without DP. We found that similarity values in the case of DP-Actor exhibited greater independence to the eligibility trace of non-DP compared with between the DP-Critic and the non-DP.

The proposed method can be applied only to a discrete state and action space. As there are many continuous spaces in the real world, in the future work, we are bound to extend our work to the more complicated continuous spaces where we can construct the value functions by high performing deep learning techniques (i.e., deep reinforcement learning). In this way, we will aim not only to solve complicated problems reflecting real world situations but also to enhance performance. In addition, we also plan to develop DP version of various state-of-the-art RL algorithms.

Author Contributions: Conceptualization, K.S.; methodology, K.S.; software, K.S.; validation, J.Y.; formal analysis, K.S.; investigation, K.S. and J.Y.; resources, J.Y.; data curation, K.S.; writing—original draft preparation, K.S.; writing—review and editing, J.Y.; supervision, J.Y.; project administration, J.Y.; funding acquisition, J.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (Grant No. 2018R1D1A1B07048790).

Acknowledgments: We thank Jonanthan Lebensold for providing the Markov Decision Process experiment code and his experiment's results.

Conflicts of Interest: The authors declare no conflict of interest.

References

- 1. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed.; The MIT Press: Cambridge, MA, USA, 2018.
- 2. Dwork, C. Differential Privacy. In *Lecture Notes in Computer Science*; Automata, Languages and Programming (ICALP 2006), Part II; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4052, pp. 1–12.
- 3. Abadi, M.; Chu, A.; Goodfellow, I.J.; McMahan, H.B.; Mironov, I.; Talwar, K.; Zhang, L. Deep Learning with Differential Privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 308-318.
- 4. Balle, B.; Gomrokchi, M.; Precup, D. Differentially Private Policy Evaluation. In *JMLR Workshop and Conference Proceedings*; Balcan, M.F., Weinberger, K.Q., PMLR: New York City, NY, USA, 2016; Volume 48, pp. 2130–2138.
- 5. Lebensold, J.; Hamilton, W.; Balle, B.; Precup, D. Actor Critic with Differentially Private Critic. *arXiv* **2019**, arXiv:1910.05876.
- 6. Konda, V.R.; Tsitsiklis, J.N. Actor-Critic Algorithms. In *Advances in Neural Information Processing Systems* 12; Solla, S.A., Leen, T.K., Müller, K., Eds.; MIT Press: Cambridge, MA, USA, 2000; pp. 1008–1014.
- 7. Konda, V.R.; Tsitsiklis, J.N. On Actor-Critic Algorithms. SIAM J. Control Optim. 2003, 42, 1143–1166. [CrossRef]
- 8. Dwork, C.; McSherry, F.; Nissim, K.; Smith, A. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography, TCC 2006, Lecture Notes in Computer Science*; TCC, Halevi, S., Rabin, T., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; Volume 3876, pp. 265–284.
- 9. Xie, T.; Thomas, P.S.; Miklau, G. Privacy Preserving Off-Policy Evaluation. arXiv 2019, arXiv:1902.00174.
- 10. Song, S.; Chaudhuri, K.; Sarwate, A.D. Stochastic Gradient Descent with Differentially Private Updates. In Proceedings of the 2013 IEEE Global Conference on Signal and Information Processing, Austin, TX, USA, 3–5 December 2013; pp. 245–248.
- 11. Chaudhuri, K.; Monteleoni, C.; Sarwate, A.D. Differentially Private Empirical Risk Minimization. *J. Mach. Learn. Res.* **2011**, *12*, 1069–1109. [PubMed]
- 12. Sutton, R.S.; Maei, H.R.; Precup, D.; Bhatnagar, S.; Silver, D.; Szepesvári, C.; Wiewiora, E. Fast Gradient-Descent Methods for Temporal-Difference Learning with Linear Function Approximation. In *Proceedings of the 26th Annual International Conference on Machine Learning*; ACM: New York, NY, USA, 2009; pp. 993–1000.
- 13. Wang, B.; Hegde, N. Privacy-Preserving Q-Learning with Functional Noise in Continuous Spaces. In *Advances in Neural Information Processing Systems*; NeurIPS: Vancouver, BC, Canada, 2019; pp. 11323–11333.
- 14. Sutton, R.S. Temporal Credit Assignment in Reinforcement Learning. Ph.D. Thesis, University of Massachusetts, Amherst, MA, USA, 1984.
- 15. Degris, T.; White, M.; Sutton, R.S. Linear Off-Policy Actor-Critic. arXiv 2012, arXiv:1205.4839.
- 16. Dwork, C. Differential Privacy: A Survey of Results. In *Theory and Applications of Models of Computation, TAMC 2008, Lecture Notes in Computer Science*; TAMC, Agrawal M., Du D., Duan Z., Li A., Eds.; Springer: Berlin/Heidelberg, Germany, 2008; Volume 4978, pp. 1–19.
- 17. Maei, H.R. Gradient Temporal-Difference Learning Algorithms. Ph.D. Thesis, University of Alberta, Edmonton, AB, Canada, 2011.

18. Komorowski, M.; Celi, L.A.; Badawi, O.; Gordon, A.C.; Faisal, A.A. The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care. *Nat. Med.* **2018**, *24*, 1716–1720. [CrossRef] [PubMed]

19. Dietterich, T.G. Hierarchical reinforcement learning with the MAXQ value function decomposition. J. Artif. Intell. Res. 2000, 13, 227–303. [CrossRef]



 \odot 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).