Differentially Private Policy Evaluation

Borja Balle B.DEBALLEPIGEM@LANCASTER.AC.UK
Lancaster University

Maziar Gomrokchi MGOMRO@CS.MCGILL.CA
Doina Precup DPRECUP@CS.MCGILL.CA
McGill University

Abstract

We present the first differentially private algorithms for reinforcement learning, which apply to the task of evaluating a fixed policy. We establish two approaches for achieving differential privacy, provide a theoretical analysis of the privacy and utility of the two algorithms, and show promising results on simple empirical examples.

1. Introduction

Learning how to make decisions under uncertainty is becoming paramount in many practical applications, such as medical treatment design, energy management, adaptive user interfaces, recommender systems etc. Reinforcement learning (Sutton & Barto, 1998) provides a variety of algorithms capable of handling such tasks. However, in many practical applications, aside from obtaining good predictive performance, one might also require that the data used to learn the predictor be kept confidential. This is especially true in medical applications, where patient confidentiality is very important, and in other applications which are user-centric (such as recommender systems). *Differential privacy* (DP) (Dwork, 2006) is a very active research area, originating from cryptography, but which has now been embraced by the machine learning community. DP is a formal model of privacy used to design mechanisms that reduce the amount of information leaked by the result of queries to a database containing sensitive information about multiple users (Dwork, 2006). Many supervised learning algorithms have differentially private versions, including logistic regression (Chaudhuri & Monteleoni, 2009; Chaudhuri et al., 2011), support vector machines (Chaudhuri et al., 2011; Rubinstein et al., 2012; Jain & Thakurta, 2013), and the lasso (Thakurta & Smith, 2013). However, differential privacy for reinforcement learning tasks has not been tackled yet, except for the simpler case of bandit problems (Smith & Thakurta, 2013; Mishra & Thakurta, 2015; Tossou & Dimitrakakis, 2016).

In this paper, we tackle differential privacy for reinforcement learning algorithms for the full Markov Decision Process (MDP) setting. We develop differentially private algorithms for the problem of policy evaluation, in which a given way of behaving has to be evaluated quantitatively. We start with the batch, first-visit Monte Carlo approach to policy evaluation, which is well understood and closest to regression algorithms, and provide two differentially private versions, which come with formal privacy proofs as well as guarantees on the quality of the solution obtained. Both algorithms work by injecting Gaussian noise into the parameters vector for the value functions, but they differ in the definition of the noise amount. Our privacy analysis techniques are related to previous output perturbation for empirical risk minimization (ERM), but there are some domain specific challenges that need to be addressed. In particular, the notion of neighbouring datasets we use is motivated by medical applications where individual patients generate full trajectories. In this case two datasets differing in a single patient yield regression problems differing in multiple correlated regression targets. Our utility analysis identifies parameters of the MDP that control how easy it is to maintain privacy in each case. The theoretical utility analysis, as well as some illustrative experiments, show that the accuracy of the private algorithms does not suffer (compared to usual Monte Carlo) when the data set is large.

The rest of the paper is organized as follows. In Sec. 2 we provide background notation and results on differential privacy and Monte Carlo methods for policy evaluation. Sec. 3 presents our proposed algorithms. The privacy analysis and the utility analysis are outlined in Sec. 4 and Sec. 5 respectively. Detailed proofs for both of these sections are given in the Supplementary Material. In Sec. 6 we provide empirical illustrations of the scaling behaviour of

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48. Copyright 2016 by the author(s).

the proposal algorithms, using synthetic MDPs, which try to mimic characteristics of real applications. Finally, we conclude in Sec. 7 with a discussion of related work and avenues for future work.

2. Background

2.1. Differential Privacy

DP takes a user-centric approach, by providing privacy guarantees based on the difference of the outputs of a learning algorithm trained on two databases differing in a single user. The central goal is to bound the loss in privacy that a user can suffer when the result of an analysis on a database with her data is made public. This can incentivize users to participate in studies using sensitive data, e.g. mining of medical records. In the context of machine learning, differentially private algorithms are useful because they allow learning models in such a way that their parameters do not reveal information about the training data (McSherry & Talwar, 2007). For example, one can think of using historical medical records to learn prognostic and diagnostic models which can then be shared between multiple health service providers without compromising the privacy of the patients whose data was used to train the model.

To formalize the above discussion, let $\mathcal{X}$ be an *input space* and $\mathcal{Y}$ an *output space*. Suppose A is a randomized algorithm that takes as input a tuple $X = (x_1, \ldots, x_m)$ of elements from $\mathcal{X}$ for some $m \geq 1$ and outputs a (random) element $A(X)$ of $\mathcal{Y}$. We interpret $X \in \mathcal{X}^m$ as a dataset containing data from m individuals and define its *neighbouring* datasets as those that differ from X in their last[1] element: $X' = (x_1, \ldots, x_{m-1}, x'_m)$ with $x_m \neq x'_m$. We denote this (symmetric) relation by $X \simeq X'$. Algorithm A is (ε, δ)-*differentially private* for some $\varepsilon, \delta > 0$ if for every $m \geq 1$, every pair of datasets $X, X' \in \mathcal{X}^m$, $X \simeq X'$, and every measurable set $\Omega \subseteq \mathcal{Y}$ we have

$$\mathbb{P}[A(X) \in \Omega] \leq e^{\varepsilon} \mathbb{P}[A(X') \in \Omega] + \delta \ . \tag{1}$$

This definition means that the distribution over possible outputs of A on inputs X and X' is very similar, so revealing this output leaks almost no information on whether x_m or x'_m was in the dataset.

A simple way to design a DP algorithm for a given function $f : \mathcal{X}^m \to \mathcal{Y}$ is the *output perturbation* mechanism, which releases $A(X) = f(X) + \eta$, where η is noise sampled from a properly calibrated distribution. For real outputs $\mathcal{Y} = \mathbb{R}^d$, the Laplace (resp. Gaussian) mechanism (see e.g. Dwork & Roth (2014)) samples each component of the noise $\eta = (\eta_1, \ldots, \eta_d)$ i.i.d. from a Laplace (resp. Gaussian) distribution with standard deviation $O(\mathrm{GS}_1(f)/\varepsilon)$ (resp. $O(\mathrm{GS}_2(f)\ln(1/\delta)/\varepsilon)$), where $\mathrm{GS}_p(f)$ is the *global sensitivity* of f given by

$$\mathrm{GS}_p(f) = \sup_{X, X' \in \mathcal{X}^m, X \simeq X'} \|f(X) - f(X')\|_p \ .$$

Calibrating noise to the global sensitivity is a worst-case approach that requires taking the supremum over all possible pairs of neighbouring datasets, and in general does not account for the fact that in some datasets privacy can be achieved with substantially smaller perturbations. In fact, for many applications (like the one we consider in this paper) the global sensitivity is too large to provide useful mechanisms. Ideally one would like to add perturbations proportional to the potential changes around the input dataset X, as measured, for example by the *local sensitivity* $\mathrm{LS}_p(f, X) = \sup_{X' \simeq X} \|f(X) - f(X')\|_p$. Nissim et al. (2007) showed that approaches based on LS_p do not lead to differentially private algorithms, and then proposed an alternative framework for DP mechanisms with data-dependent perturbations based on the idea of *smoothed sensitivity*. This is the approach we use in this paper; see Section 4 for further details.

2.2. Policy Evaluation

Policy evaluation is the problem of obtaining (an approximation to) the value function of a Markov reward process defined by an MDP M and a policy π (Sutton & Barto, 1998; Szepesvári, 2010). In many cases of interest M is unknown but we have access to trajectories containing state transitions and immediate rewards sampled from π. When the state space of M is relatively small, tabular methods that represent the value of each state can be used individually. However, in problems with large (or even continuous) state spaces, parametric representations for the value function are typically needed in order to defeat the curse of dimensionality and exploit the fact that similar states will have similar values. In this paper we focus on policy evaluation with linear function approximation in the batch case, where we have access to a set of trajectories sampled from the policy of interest.

Let M be an MDP over a finite state space $\mathcal{S}$ with $N = |\mathcal{S}|$ states and π a policy on M. Given an initial state $s_0 \in \mathcal{S}$, the interaction of π with M is described by a sequence $x = ((s_t, a_t, r_t))_{t \geq 0}$ of state–action–reward triplets. Suppose $0 < \gamma < 1$ is the discount factor of M. The value function $V^\pi : \mathcal{S} \to \mathbb{R}$ of π assigns to each state the expected discounted cumulative reward obtained by a trajectory following policy π from that state:

$$V^\pi(s) = \mathbb{E}_{M,\pi}\left[\sum_{t \geq 0} \gamma^t r_t \,\middle|\, s_0 = s \right] \ . \tag{2}$$

[1] Formally, a neighbouring datasets is one which differs in one element, not necessarily the last. However, here we assume the order of the elements in X does not affect the distribution of $A(X)$, and thus define without loss of generality neighbouring datasets as always differing in the last element.

The value function can be considered a vector $V^\pi \in \mathbb{R}^{\mathcal{S}}$. We make the usual assumption that any reward r generated by M is bounded: $0 \leq r \leq R_{\max}$, so $0 \leq V^\pi(s) \leq R_{\max}/(1-\gamma)$ for all $s \in \mathcal{S}$.

Let $\Phi \in \mathbb{R}^{\mathcal{S}\times d}$ be a feature representation that associates each state $s \in \mathcal{S}$ to a d-dimensional feature vector $\phi_s^\top = \Phi(s,:) \in \mathbb{R}^d$. The goal is to find a parameter vector $\theta \in \mathbb{R}^d$ such that $\hat{V}^\pi = \Phi\theta$ is a good approximation to V^π. To do so, we assume that we have access to a collection $X = (x_1, \ldots, x_m)$ of finite trajectories sampled from M by π, where each x_i is a sequence of states, actions and rewards.

We will use a Monte Carlo approach, in which the returns of the trajectories in X are used as regression targets to fit the parameters in $\hat{V}^\pi$ via a least squares approach (Sutton & Barto, 1998). In particular, we consider first-visit Monte Carlo estimates obtained as follows. Suppose $x = ((s_1, a_1, r_1), \ldots, (s_T, a_T, r_T))$ is a trajectory that visits s and $i_{x,s}$ is the time of the first visit to s; that is, $s_{i_{x,s}} = s$, and $s_t \neq s$ for all $t < i_{x,s}$. The return collected from this first visit is given by

$$F_{x,s} = \sum_{t=i_{x,s}}^{T} r_t \gamma^{t-i_{x,s}} = \sum_{t=0}^{T-i_{x,s}} r_{t+i_{x,s}} \gamma^t \ ,$$

and provides an unbiased estimate of $V^\pi(s)$. For convenience, when state s is not visited by trajectory x we assume $F_{x,s} = 0$.

Given the returns from all first visits corresponding to a dataset X with m trajectories, we can find a parameter vector for the estimator $\hat{V}^\pi$ by solving the optimization problem $\operatorname{argmin}_\theta J_X(\theta)$, where

$$J_X(\theta) = \frac{1}{m} \sum_{i=1}^{m} \sum_{s \in \mathcal{S}_{x_i}} \rho_s (F_{x_i,s} - \phi_s^\top \theta)^2 \ , \qquad (3)$$

and $\mathcal{S}_x$ is the set of states visited by trajectory x. The regression weights $0 \leq \rho_s \leq 1$ are given as an input to the problem and capture the user's believe that some states are more relevant than others. It is obvious that $J_X(\theta)$ is a convex function of θ. However, in general it is not strongly convex and therefore the optimum of $\operatorname{argmin}_\theta J_X(\theta)$ is not necessarily unique. On the other hand, it is known that differential privacy is tightly related to certain notions of stability (Thakurta & Smith, 2013), and optimization problems with non-unique solutions generally pose a problem to stability. In order to avoid this problem, the private policy evaluation algorithms that we propose in Section 3 are based on optimizing slightly modified versions of $J_X(\theta)$ which promote stability in their solutions. Note that the notions of stability related to DP are for worst-case situations: that is, they need to hold for every possible pair of neighbouring input dataset $X \simeq X'$, regardless of any generative model assumed for the trajectories in those datasets. In particular, these stability considerations are not directly related to the variance of the estimates in $\hat{V}^\pi$.

We end this section by introducing further notation that will be used in the sequel. Given a dataset X with m trajectories let $F_X \in \mathbb{R}^{\mathcal{S}}$ denote the vector containing the average first visit returns from all trajectories in X that visit a particular state. In particular, if X_s represents the multiset of trajectories from X that visit state s at some point, then we have

$$F_X(s) = F_{X,s} = \frac{1}{|X_s|} \sum_{x \in X_s} F_{x,s} \ . \qquad (4)$$

If s is not visited by any trajectory in X we set $F_{X,s} = 0$. We also define a diagonal matrix $\Gamma_X \in \mathbb{R}^{\mathcal{S}\times\mathcal{S}}$ with entries given by the product of the regression weight on each state and the fraction of trajectories in X visiting that state: $\Gamma_X(s,s) = \rho_s |X_s|/m$. Now, a typical calculation solving for θ in $\nabla_\theta J_X(\theta) = 0$ shows that any $\theta_X \in \operatorname{argmin}_\theta J_X(\theta)$ must also be an optimum of

$$\sum_{s \in \mathcal{S}} \frac{\rho_s |X_s|}{m} (F_{X,s} - \phi_s^\top \theta)^2 \ . \qquad (5)$$

Alternatively, we can say θ_X is an optimum of $J_X(\theta)$ if and only if it satisfies

$$\Phi^\top \Gamma_X \Phi \theta_X = \Phi^\top \Gamma_X F_X \ . \qquad (6)$$

Hence, $J_X(\theta)$ has a unique global optimum if and only if the matrix $\Phi^\top \Gamma_X \Phi$ is invertible. Since it is possible to find neighbouring datasets $X \simeq X'$ where at most one of $\Phi^\top \Gamma_X \Phi$ and $\Phi^\top \Gamma_{X'} \Phi$ is invertible, using $J_X(\theta)$ to define the policy evaluation problem poses a problem to the design differentially private algorithms. Next we discuss two ways to make this optimization more stable, leading to two different DP policy evaluation algorithms.

3. Private First-Visit Monte Carlo Algorithms

In this section we give the details of two differentially private policy evaluation algorithms based on first-visit Monte Carlo estimates. A formal privacy analysis of these algorithms is given in Section 4. Bounds showing how the privacy requirement affects the utility of the value estimates are presented in Section 5.

3.1. Algorithm DP-LSW

One way to make the optimization $\operatorname{argmin}_\theta J_X(\theta)$ more stable to changes in the dataset X is to consider an alternative least-squares optimization leading to a closed form solution similar to (4) but where the invertibility of the coefficient matrix does not change with X. Thus, we modify the objective function (5) by introducing a new set of positive regression weights $w_s > 0$ and letting $\Gamma \in \mathbb{R}^{\mathcal{S}\times\mathcal{S}}$ be a

diagonal matrix with $\Gamma(s,s) = w_s$. In this way we obtain the objective function

$$J_X^w(\theta) = \sum_{s\in\mathcal{S}} w_s (F_{X,s} - \phi_s^\top \theta)^2 = \|F_X - \Phi\theta\|_{2,\Gamma}^2 \ , \quad (7)$$

where $\|v\|_{2,\Gamma}^2 = \|\Gamma^{1/2} v\|_2^2 = v^\top \Gamma v$ is a weighted L_2 norm. Using the same argument as before we see that any $\theta_X^w \in \operatorname{argmin}_\theta J_X^w(\theta)$ must satisfy

$$\Phi^\top \Gamma \Phi \theta_X^w = \Phi^\top \Gamma F_X \ . \quad (8)$$

Thus, the new optimization problem is well-posed whenever $\Phi^\top\Gamma\Phi$ is invertible, which henceforth will be our working assumption. Note that this is a mild assumption, since it is satisfied by choosing a feature matrix Φ with full column rank. Under this assumption we have:

$$\theta_X^w = \left(\Phi^\top\Gamma\Phi\right)^{-1} \Phi^\top \Gamma F_X = \left(\Gamma^{1/2}\Phi\right)^\dagger \Gamma^{1/2} F_X \ , \quad (9)$$

where $M^\dagger$ denotes the Moore–Penrose pseudo-inverse. The difference between optimizing $J_X(\theta)$ or $J_X^w(\theta)$ is reflected in the differences between (6) and (8). In particular, if the trajectories in X are i.i.d. and p_s denotes the probability that state s is visited by a trajectory in X, then taking $w_s = \mathbb{E}_X[\rho_s |X_s|/m] = \rho_s p_s$ yields a loss function $J_X^w(\theta)$ that captures the effect of each state s in $J_X(\theta)$ in the asymptotic regime $m \to \infty$. However, we note that knowledge of these visit probabilities is not required for running our algorithm or for its analysis.

Our first DP algorithm for policy evaluation applies a carefully calibrated output perturbation mechanism to the solution θ_X^w of $\operatorname{argmin}_\theta J_X^w(\theta)$. This algorithm is called DP-LSW and its full pseudo-code is given in Algorithm 1. It receives as input the dataset X, the regression weights w, the feature representation Φ, and the MDP parameters $R_{\max}$ and γ. Additionally, the algorithm is parametrized by the privacy parameters ε and δ. Its output is the result of adding a random vector η drawn from a multivariate Gaussian distribution $\mathcal{N}(0, \sigma_X^2 I)$ to the parameter vector θ_X^w. In order to compute the variance of η the algorithm needs to solve the discrete optimization problem $\psi_X^w = \max_{0\le k\le K_X} e^{-k\beta}\varphi_X^w(k)$, where $K_X = \max_{s\in\mathcal{S}} |X_s|$, β is a parameter computed in the algorithm, and $\varphi_X^w(k)$ is given by the following expression:

$$\varphi_X^w(k) = \sum_{s\in\mathcal{S}} \frac{w_s}{\max\{|X_s| - k, 1\}^2} \ . \quad (10)$$

Note that ψ_X^w can be computed in time $O(K_X N)$.

The variance of the noise in DP-LSW is proportional to the upper bound $R_{\max}/(1-\gamma)$ on the return from any state. This bound might be excessively pessimistic in some applications, leading to unnecessary large perturbation of the solution θ_X^w. Fortunately, it is possible to replace the term $R_{\max}/(1-\gamma)$ with any smaller upper bound $F_{\max}$ on the returns generated by the target MDP on any state. In practice this leads to more useful algorithms, but it is important to keep in mind that for the privacy guarantees to remain unaffected, one needs to assume that $F_{\max}$ is a publicly known quantity (i.e. it is not based on an estimate made from private data). These same considerations apply to the algorithm in the next section.

Algorithm 1: DP-LSW

Input: X, Φ, γ, $R_{\max}$, w, ε, δ
Output: $\hat{\theta}_X^w$

Compute θ_X^w ; // cf. (9)
Let $\alpha \leftarrow \frac{5\sqrt{2\ln(2/\delta)}}{\varepsilon}$ and $\beta \leftarrow \frac{\varepsilon}{4(d+\ln(2/\delta))}$;
Let $\psi_X^w \leftarrow \max_{0\le k\le K_X} e^{-k\beta}\varphi_X^w(k)$; // cf. (10)
Let $\sigma_X \leftarrow \frac{\alpha R_{\max}\|(\Gamma^{1/2}\Phi)^\dagger\|}{1-\gamma}\sqrt{\psi_X^w}$;
Sample a d-dimensional vector $\eta \sim \mathcal{N}(0, \sigma_X^2 I)$;
Return $\hat{\theta}_X^w = \theta_X^w + \eta$;

3.2. Algorithm DP-LSL

The second DP algorithm for policy evaluation we propose is also an output perturbation mechanism. It differs from DP-LSW in they way stability of the unperturbed solutions is promoted. In this case, we choose to optimize a regularized version of $J_X(\theta)$. In particular, we consider the objective function $J_X^\lambda(\theta)$ obtained by adding a ridge penalty to the least-squares loss from (3):

$$J_X^\lambda(\theta) = J_X(\theta) + \frac{\lambda}{2m}\|\theta\|_2^2 \ , \quad (11)$$

where $\lambda > 0$ is a regularization parameter. The introduction of the ridge penalty makes the objective function $J_X^\lambda(\theta)$ strongly convex, and thus ensures the existence of a unique solution $\theta_X^\lambda = \operatorname{argmin}_\theta J_X^\lambda(\theta)$, which can be obtained in closed-form as:

$$\theta_X^\lambda = \left(\Phi^\top\Gamma_X\Phi + \frac{\lambda}{2m} I\right)^{-1} \Phi^\top \Gamma_X F_X \ . \quad (12)$$

Here Γ_X is defined as in Section 2.2.

We call DP-LSL the algorithm obtained by applying an output perturbation mechanism to the minimizer of $J_X^\lambda(\theta)$; the full pseudo-code is given in Algorithm 2. It receives as input the privacy parameters ε and δ, a dataset of trajectories X, the regression weights ρ, the feature representation Φ, a regularization parameter $\lambda > \|\Phi\|^2\|\rho\|_\infty$, and the MDP parameters $R_{\max}$ and γ. After computing the solution θ_X^λ to $\operatorname{argmin}_\theta J_X^\lambda(\theta)$, the algorithm outputs

$\hat{\theta}^\lambda_X = \theta^\lambda_X + \eta$, where η is a d-dimensional noise vector drawn from $\mathcal{N}(0, \sigma^2_X I)$. The variance of η is obtained by solving a discrete optimization problem (different from the one in DP-LSW). Let $c_\lambda = \|\Phi\|\|\rho\|_\infty/\sqrt{2\lambda}$ and for $k \geq 0$, define $\varphi^\lambda_X(k)$ as:

$$\left(c_\lambda \sqrt{\sum_s \rho_s \min\{|X_s| + k, m\}} + \|\rho\|_2 \right)^2 . \quad (13)$$

Then DP-LSL computes $\psi^\lambda_X = \max_{0 \leq k \leq m} e^{-k\beta} \varphi^\lambda_X(k)$, which can be done in time $O(mN)$.

Algorithm 2: DP-LSL

Input: X, Φ, γ, $R_{\max}$, ρ, λ, ε, δ
Output: $\hat{\theta}^\lambda_X$
Compute θ^λ_X ; // cf. (12)
Let $\alpha \leftarrow \frac{5\sqrt{2\ln(2/\delta)}}{\varepsilon}$ and $\beta \leftarrow \frac{\varepsilon}{4(d+\ln(2/\delta))}$;
Let $\psi^\lambda_X \leftarrow \max_{0 \leq k \leq m} e^{-k\beta} \varphi^\lambda_X(k)$; // cf. (13)
Let $\sigma_X \leftarrow \frac{2\alpha R_{\max}\|\Phi\|}{(1-\gamma)(\lambda - \|\Phi\|^2\|\rho\|_\infty)} \sqrt{\psi^\lambda_X}$;
Sample a d-dimensional vector $\eta \sim \mathcal{N}(0, \sigma^2_X I)$;
Return $\hat{\theta}^\lambda_X = \theta^\lambda_X + \eta$;

4. Privacy Analysis

This section provides a formal privacy analysis for DP-LSW and DP-LSL and shows that both algorithms are (ε, δ)-differentially private. We use the smooth sensitivity framework of (Nissim et al., 2007; 2011), which provides tools for the design of DP mechanisms with data-dependent output perturbations. We rely on the following lemma, which provides sufficient conditions for calibrating Gaussian output perturbation mechanisms with variance proportional to smooth upper bounds of the local sensitivity.

Lemma 1 (Nissim et al. (2011))**.** *Let A be an algorithm that on input X computes a vector $\mu_X \in \mathbb{R}^d$ deterministically and then outputs $Z_X \sim \mathcal{N}(\mu_X, \sigma^2_X I)$, where σ^2_X is a variance that depends on X. Let $\alpha = \alpha(\varepsilon, \delta) = 5\sqrt{2\ln(2/\delta)}/\varepsilon$ and $\beta = \beta(\varepsilon, \delta, d) = \varepsilon/(4d + 4\ln(2/\delta))$. Suppose ε and δ are such that the following are satisfied for every pair of neighbouring datasets $X \simeq X'$: (a) $\sigma_X \geq \alpha\|\mu_X - \mu_{X'}\|_2$, and (b) $|\ln(\sigma^2_X) - \ln(\sigma^2_{X'})| \leq \beta$. Then A is (ε, δ)-differentially private.*

Condition (a) says we need variance at least proportional to the local sensitivity $\mathrm{LS}_2(f, X)$. Condition (b) asks that the variance does not change too fast between neighbouring datasets by imposing the constraint $\sigma^2_X/\sigma^2_{X'} \leq e^\beta$. This is precisely the spirit of the smoothed sensitivity principle: calibrate the noise to a smooth upper bound of the local sensitivity. We acknowledge Lemma 1 is only available in pre-print form, and thus provide an elementary proof in the Supplementary Material for completeness. The remaining proofs from this section are also presented there.

4.1. Privacy Analysis of DP-LSW

We start by providing an upper bound on the norm $\|\theta^w_X - \theta^w_{X'}\|_2$ for any two neighbouring datasets $X \simeq X'$. Using (9) it is immediate that:

$$\|\theta^w_X - \theta^w_{X'}\|_2 \leq \|(\Gamma^{1/2}\Phi)^\dagger\| \|F_X - F_{X'}\|_{2,\Gamma} . \quad (14)$$

Next we provide an upper bound to $\|F_X - F_{X'}\|_{2,\Gamma}$.

Lemma 2. *Let $X \simeq X'$ be two neighbouring datasets of m trajectories with $X = (x_1, \ldots, x_{m-1}, x)$ and $X' = (x_1, \ldots, x_{m-1}, x')$. Let $X^\circ = (x_1, \ldots, x_{m-1})$. Let $\mathcal{S}_x$ (resp. $\mathcal{S}_{x'}$) denote the set of states visited by x (resp. x'). Then we have*

$$\|F_X - F_{X'}\|_{2,\Gamma} \leq \frac{R_{\max}}{1-\gamma} \sqrt{\sum_{s \in \mathcal{S}_x \cup \mathcal{S}_{x'}} \frac{w_s}{(|X^\circ_s| + 1)^2}} .$$

Since the condition in Lemma 1 needs to hold for any dataset X' neighbouring X, we take the supremum of the bound above over all neighbours, which yields the following corollary.

Corollary 3. *If X is a dataset of trajectories, then the following holds for every neighbouring dataset $X' \simeq X$:*

$$\|F_X - F_{X'}\|_{2,\Gamma} \leq \frac{R_{\max}}{1-\gamma} \sqrt{\sum_{s \in \mathcal{S}} \frac{w_s}{\max\{|X_s|, 1\}^2}} .$$

Using this result we see that in order to satisfy item (a) of Lemma 1 we can choose a noise variance satisfying:

$$\sigma_X \geq \frac{\alpha R_{\max}\|(\Gamma^{1/2}\Phi)^\dagger\|}{1-\gamma} \sqrt{\sum_{s \in \mathcal{S}} \frac{w_s}{\max\{|X_s|, 1\}^2}} , \quad (15)$$

where only the last multiplicative term depends on the dataset X, and the rest can be regarded as a constant that depends on parameters of the problem which are either public or chosen by the user, and will not change for a neighbouring dataset X'. Thus, we are left with a lower bound expressible as $\sigma_X \geq C\sqrt{\varphi^w_X}$, where $\varphi^w_X = \sum_s (w_s/\max\{|X_s|, 1\}^2)$ only depends on the dataset X through its *signature* $\langle X \rangle \in \mathbb{N}^{\mathcal{S}}$ given by the number of times each state appears in the trajectories of X: $\langle X \rangle(s) = |X_s|$. Accordingly, we write $\varphi^w_X = \varphi^w(\langle X \rangle)$, where $\varphi^w : \mathbb{N}^{\mathcal{S}} \to \mathbb{R}$ is the function

$$\varphi^w(v) = \sum_s \frac{w_s}{\max\{v_s, 1\}^2} . \quad (16)$$

The signatures of two neighbouring datasets $X \simeq X'$ satisfy $\|\langle X \rangle - \langle X' \rangle\|_\infty \leq 1$ because replacing a single trajectory can only change by one the number of first visits to any particular state. Thus, assuming we have a

function $\psi : \mathbb{N}^{\mathcal{S}} \to \mathbb{R}$ satisfying $\psi^w(v) \geq \varphi^w(v)$ and $|\ln(\psi^w(v)) - \ln(\psi^w(v'))| \leq \beta$ for all $v, v' \in \mathbb{N}^{\mathcal{S}}$ with $\|v - v'\|_\infty \leq 1$, we can take $\sigma_X = C\sqrt{\psi^w(\langle X \rangle)}$. This variance clearly satisfies the conditions of Lemma 1 since

$$|\ln(\sigma_X^2) - \ln(\sigma_{X'}^2)| = |\ln(\psi^w(\langle X \rangle)) - \ln(\psi^w(\langle X' \rangle))| \leq \beta \ .$$

The function ψ^w is known as a *β-smooth upper bound* of φ^w, and the following result provides a tool for constructing such functions.

Lemma 4 (Nissim et al. (2007)). *Let $\varphi : \mathbb{N}^{\mathcal{S}} \to \mathbb{R}$. For any $k \geq 0$ let $\varphi_k(v) = \max_{\|v - v'\|_\infty \leq k} \varphi(v')$. Given $\beta > 0$, the smallest β-smooth upper bound of φ is the function*

$$\psi(v) = \sup_{k \geq 0} \left(e^{-k\beta} \varphi_k(v) \right) \ . \tag{17}$$

For some functions φ, the upper bound ψ can be hard to compute or even approximate (Nissim et al., 2007). Fortunately, in our case a simple inspection of (16) reveals that $\varphi_k^w(v)$ is easy to compute. In particular, the following lemma implies that $\psi^w(v)$ can be obtained in time $O(N\|v\|_\infty)$.

Lemma 5. *The following holds for every $v \in \mathbb{N}^{\mathcal{S}}$:*

$$\varphi_k^w(v) = \sum_{s \in \mathcal{S}} \frac{w_s}{\max\{v_s - k, 1\}^2} \ .$$

Furthermore, for every $k \geq \|v\|_\infty - 1$ we have $\varphi_k^w(v) = \sum_s w_s$.

Combining the last two lemmas, we see that the quantity ψ_X^w computed in DP-LSW is in fact a β-smooth upper bound to φ_X^w. Because the variance σ_X used in DP-LSW can be obtained by plugging this upper bound into (15), the two conditions of Lemma 1 are satisfied. This completes the proof of the main result of this section:

Theorem 6. *Algorithm DP-LSW is (ε, δ)-differentially private.*

Before proceeding to the next privacy analysis, note that Corollary 3 is the reason why a mechanism with output perturbations proportional to the global sensitivity is not sufficient in this case. The bound there says that if in the worst case we can find datasets of an arbitrary size m where some states are visited few (or zero) times, then the global sensitivity will not vanish as $m \to \infty$. Hence, the utility of such algorithm would not improve with the size of the dataset. The smoothed sensitivity approach works around this problem by adding large noise to these datasets, but adding much less noise to datasets where each state appears a sufficient number of times. Corollary 3 also provides the basis for efficiently computing smooth upper bounds to the local sensitivity. In principle, condition (b) in Lemma 1 refers to any dataset neighbouring X, of which there are uncountably many because we consider real rewards. Bounding the local sensitivity in terms of the signature reduces this to finitely many "classes" of neighbours, and the form of the bound in Corollary 3 makes it possible to apply Lemma 4 efficiently.

4.2. Privacy Analysis of DP-LSL

The proof that DP-LSL is differentially private follows the same strategy as for DP-LSW. We start with a lemma that bounds the local sensitivity of θ_X^λ for pairs of neighbouring datasets $X \simeq X'$. We use the notation $\mathbb{I}_{s \in x}$ for an indicator variable that is equal to one when state s is visited within trajectory x.

Lemma 7. *Let $X \simeq X'$ be two neighbouring datasets of m trajectories with $X = (x_1, \ldots, x_{m-1}, x)$ and $X' = (x_1, \ldots, x_{m-1}, x')$. Let $F_x \in \mathbb{R}^{\mathcal{S}}$ (resp. $F_{x'} \in \mathbb{R}^{\mathcal{S}}$) be the vector given by $F_x(s) = F_{x,s}$ (resp. $F_{x'}(s) = F_{x',s}$). Define diagonal matrices $\Gamma_\rho, \Delta_{x,x'} \in \mathbb{R}^{\mathcal{S} \times \mathcal{S}}$ given by $\Gamma_\rho(s,s) = \rho_s$ and $\Delta_{x,x'}(s,s) = \mathbb{I}_{s \in x} - \mathbb{I}_{s \in x'}$. If the regularization parameter satisfies $\lambda > \|\Phi^\top \Delta_{x,x'} \Gamma_\rho \Phi\|$, then:*

$$\frac{\|\theta_X^\lambda - \theta_{X'}^\lambda\|_2}{2} \leq \frac{\left\| \left(\Delta_{x,x'} \Phi \theta_X^\lambda - F_x + F_{x'} \right)^\top \Gamma_\rho \Phi \right\|_2}{\lambda - \|\Phi^\top \Delta_{x,x'} \Gamma_\rho \Phi\|} \ .$$

As before, we need to consider the supremum of the bound over all possible neighbours X' of X. In particular, we would like to get a bound whose only dependence on the dataset X is through the signature $\langle X \rangle$. This is the purpose of the following corollary:

Corollary 8. *Let X be a dataset of trajectories and suppose $\lambda > \|\Phi\|^2 \|\rho\|_\infty$. Then the following holds for every neighbouring dataset $X' \simeq X$:*

$$\|\theta_X^\lambda - \theta_{X'}^\lambda\|_2 \leq \frac{2R_{\max}\|\Phi\|}{(1-\gamma)(\lambda - \|\Phi\|^2\|\rho\|_\infty)} \sqrt{\varphi_X^\lambda} \ ,$$

where

$$\varphi_X^\lambda = \left(\frac{\|\Phi\| \|\rho\|_\infty}{\sqrt{2\lambda}} \sqrt{\sum_{s \in \mathcal{S}} \rho_s |X_s|} + \|\rho\|_2 \right)^2 \ .$$

By the same reasoning of Section 4.1, as long as the regularization parameter is larger than $\|\Phi\|^2\|\rho\|_\infty$, a differentially private algorithm can be obtained by adding to θ_X^λ a Gaussian perturbation with a variance satisfying

$$\sigma_X \geq \frac{2\alpha R_{\max}\|\Phi\|}{(1-\gamma)(\lambda - \|\Phi\|^2\|\rho\|_\infty)} \sqrt{\varphi_X^\lambda}$$

and the second condition of Lemma 1. This second requirement can be achieved by computing a β-smooth upper bound of the function $\varphi^\lambda : \mathbb{N}^{\mathcal{S}} \to \mathbb{R}$ given by

$$\varphi^\lambda(v) = \left(\frac{\|\Phi\| \|\rho\|_\infty}{\sqrt{2\lambda}} \sqrt{\sum_{s \in \mathcal{S}} \rho_s \max\{v_s, m\}} + \|\rho\|_2 \right)^2 \ .$$

When going from φ^λ_X to $\varphi^\lambda(v)$ we substituted $|X_s|$ by $\max\{v_s, m\}$ to reflect the fact that any state cannot be visited by more than m trajectories in a dataset X of size m. It turns out that in this case the function $\varphi^\lambda_k(v) = \max_{\|v-v'\|_\infty \leq k} \varphi^\lambda(v')$ arising in Lemma 4 is also easy to compute.

Lemma 9. *For every $v \in \mathbb{N}^{\mathcal{S}}$, $\varphi^\lambda_k(v)$ is equal to:*

$$\left(\frac{\|\Phi\| \|\rho\|_\infty}{\sqrt{2\lambda}} \sqrt{\sum_{s \in \mathcal{S}} \rho_s \max\{v_s + k, m\}} + \|\rho\|_2 \right)^2 .$$

Furthermore, for every $k \geq m - \min_s v_s$ we have $\varphi^\lambda_k(v) = \left(\frac{\|\Phi\| \|\rho\|_\infty \sqrt{m}}{\sqrt{2\lambda}} \sqrt{\sum_{s \in \mathcal{S}} \rho_s} + \|\rho\|_2 \right)^2$.

Finally, in view of Lemma 4, Corollary 8, and Lemma 9, the variance of the noise perturbation in DP-LSL satisfies the conditions of Lemma 1, so we have proved the following.

Theorem 10. *Algorithm DP-LSL is (ε, δ)-differentially private.*

5. Utility Analysis

Because the promise of differential privacy has to hold for any possible pair of neighbouring datasets $X \simeq X'$, the analysis in previous section does not assume any generative model for the input dataset X. However, in practical applications we expect $X = (x_1, \ldots, x_m)$ to contain multiple trajectories sampled from the same policy on the same MDP. The purpose of this section is to show that when the trajectories x_i are i.i.d. the utility of our differentially private algorithms increases as $m \to \infty$. In other words, when the input dataset grows, the amount of noise added by our algorithms decreases, thus leading to more accurate estimates of the value function. This matches the intuition that using data from more users to estimate a fixed number f parameters leads to a smaller individual contributions from each user, and makes the privacy constraint easier to satisfy.

To measure the utility of our DP algorithms we shall bound the difference in empirical risk between the private and non-private parameters learned from a given dataset. That is, we want to show that the quantity $\mathbb{E}_{X,\eta}[J^\bullet_X(\hat{\theta}^\bullet_X) - J^\bullet_X(\theta^\bullet_X)]$ vanishes as $|X| = m \to \infty$, for both $\bullet = w$ and $\bullet = \lambda$. Due to space reasons, here we only state the main corollaries of our analysis and defer full statements and proofs to the Supplementary Material.

In the case of DP-LSW, we give bounds on the excess empirical risk that decrease quadratically with m under the assumption that either all states are visited with non-zero probability or the user sets the regression weights so that such states do not contribute to θ^w_X.

Corollary 11. *Let $\mathcal{S}_0 = \{s \in \mathcal{S} | p_s = 0\}$. If $w_s = 0$ for all $s \in \mathcal{S}_0$, then $\mathbb{E}_{X,\eta}[J^w_X(\hat{\theta}^w_X) - J^w_X(\theta^w_X)] = O(1/m^2)$.*

Our main statement for DP-LSL has a similar form, but in this case the rate of convergence depends on the choice of regularization parameter λ. In particular, we assume in the following statement that λ grows with m, and see what tensions arise int he selection of an adequate regularization schedule.

Corollary 12. *Suppose $\lambda = \omega(1)$ with respect to m. Then we have $\mathbb{E}_{X,\eta}[J^\lambda_X(\hat{\theta}^\lambda_X) - J^\lambda_X(\theta^\lambda_X)] = O(1/\lambda m + 1/\lambda^2 + m/\lambda^3)$.*

Note that taking $\lambda = \Theta(m)$ we get a bound on the excess risk of order $O(1/m^2)$. However, if we want the regularization term in $J^\lambda_X(\theta)$ to vanish as $m \to \infty$ we need $\lambda = o(m)$. We shall see importance of this trade-off in our experiments.

6. Experiments

In this section we illustrate the behaviour of the proposed algorithms on synthetic examples. The domain we use consists of a chain of N states, where in each state the agent has some probability p of staying and probability $(1 - p)$ of advancing to its right. There is a reward of 1 when the agent reaches the final, absorbing state, and 0 for all other states. While this is a toy example, it illustrates the typical case of policy evaluation in the medical domain, where patients tend to progress through stages of recovery at different speeds, and past states are not typically revisited (partly because in the medical domain, states contain historic information about past treatments). Trajectories are drawn by starting in an initial state distribution and generating state-action-reward transitions according to the described probabilities until the absorbing state is reached. Trajectories are harvested in a batch, and the same batches are processed by all algorithms.

We experiment with both a tabular representation of the value function, as well as with function approximation. In the latter case, we simply aggregate pairs of adjacent states, which are hence forced to take the same value. We compared the proposed private algorithms DP-LSW and DP-LSL with their non-private equivalents LSW and LSL. The performance measure used is average root mean squared error over the state space. The error is obtained by comparing the state values estimated by the learning algorithms against the exact values obtained by exact, tabular dynamic programming. Standard errors computed over 20 independent runs are included.

The main results are summarized in Fig. 1, for an environment with $N = 40$ states, $p = 0.5$, discount $\gamma = 0.99$, and for the DP algorithms, $\varepsilon = 0.1$ and $\delta = 0.1$. In general,

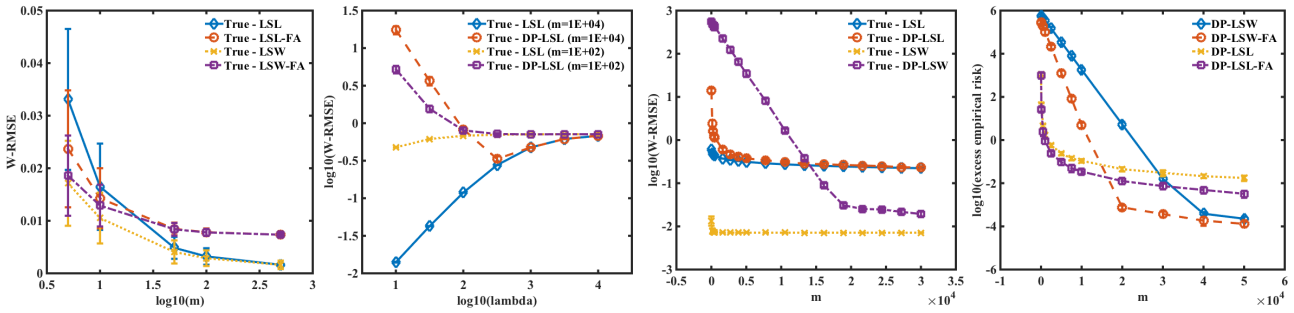

Figure 1. Empirical comparison of differentially private and non-private algorithms

these constants should be chosen depending on the privacy constraints of the domain. Our theoretical results explain the expected effect of these choices on the privacy-utility trade-off so we do not provide extensive experiments with different values.

The left plot in Fig. 1 compares the non-private LSL and LSW versions of Monte Carlo evaluation, in the tabular and function approximation case. As can be seen, both algorithms are very stable and converge to the same solution, but LSW converges faster. The second plot compares the performance of all algorithms in the tabular case, over a range of regularization parameters, for two different batch sizes. The third plot compares the expected RMSE of the algorithms when run with state aggregation, as a function of batch size. As can be seen, the DP algorithms converge to the same solutions as the non-private corresponding versions for large enough batch sizes. Interestingly, the two proposed approaches serve different needs. The LSL algorithms work better with small batches of data, whereas the LSW approach is preferable with large batches. From an empirical point of view, the trade-off between accuracy and privacy in the DP-LSL algorithm should be done by setting a regularization schedule proportional to $\sqrt{m}$. While the theory suggests it is not the best schedule in terms of excess empirical risk, it achieves the best overall accuracy.

Finally, the last plot shows excess risk as a function of the batch size. Interestingly, more aggressive function approximation helps both differentially private algorithms converge faster. This is intuitive, since using the same data to estimate fewer parameters means the effect of each individual trajectory is already obscured by the function approximation. Decreasing the number of parameters d of the function approximator increases β and lowers the smooth sensitivity bounds. In medical applications, one expects to have many attributes measured about patients, and to need aggressive function approximation in order to provide generalization. This result tells us that differentially private algorithms should be favoured in this case as well.

Overall, the empirical results are very promising, showing that especially as batch size increases, the noise introduced by the DP mechanism decreases rapidly, and these algorithms provide the same performance but with the additional privacy guarantees.

7. Conclusion

We present the first differentially private algorithms for policy evaluation in the full MDP setting. Our algorithms are built on top of established Monte Carlo methods, and come with utility guarantees showing that the cost of privacy diminishes as training batches get larger. The smoothed sensitivity framework is a key component of our analyses, which differ from previous works on DP mechanisms for ERM and bandits problems in two substantial ways. The first, we consider optimizations with non-Lipschitz loss functions, which prevents us from using most of the established techniques for analyzing privacy and utility in ERM algorithms and complicates some parts of our analysis. In particular, we cannot leverage the tight utility analysis of (Jain & Thakurta, 2014) to get dimension independent bounds. Second, and more importantly, the natural model of neighbouring datasets for policy evaluation involves replacing a whole trajectory. This implies that neighbouring datasets can differ in multiple regression targets, which is quite different from the usual supervised learning approach where neighbouring datasets can only change a single regression target. Our approach is also different from the on-line learning and bandits setting, where there is a single stream of experience and neighbouring datasets differ in one element of the stream. Note that this setting cannot be used naturally in the full MDP setup, because successive observations in a single stream are inherently correlated.

In future work we plan to extend our techniques in two directions. First, we would like to design DP policy evaluation methods based on temporal-difference learning (Sutton, 1988). Secondly, we will tackle the control case, where policy evaluation is often used as a sub-routine, e.g. as in actor-critic methods. We also plan to evaluate the current algorithms on patient data from an ongoing clinical study (in which case, errors cannot be estimated precisely, because the right answer is not known).

References

Chaudhuri, Kamalika and Monteleoni, Claire. Privacy-preserving logistic regression. In *Advances in Neural Information Processing Systems*, pp. 289–296, 2009.

Chaudhuri, Kamalika, Monteleoni, Claire, and Sarwate, Anand D. Differentially private empirical risk minimization. volume 12. JMLR. org, 2011.

Dwork, Cynthia. Differential privacy. In *Proceedings of the 33rd international conference on Automata, Languages and Programming-Volume Part II*, pp. 1–12, 2006.

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

Jain, Prateek and Thakurta, Abhradeep. Differentially private learning with kernels. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pp. 118–126, 2013.

Jain, Prateek and Thakurta, Abhradeep Guha. (near) dimension independent risk bounds for differentially private learning. In *Proceedings of The 31st International Conference on Machine Learning*, pp. 476–484, 2014.

McSherry, Frank and Talwar, Kunal. Mechanism design via differential privacy. In *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*, pp. 94–103. IEEE, 2007.

Mishra, Nikita and Thakurta, Abhradeep. Nearly optimal differentially private stochastic multi-arm bandits. In *UAI*, 2015.

Nissim, Kobbi, Raskhodnikova, Sofya, and Smith, Adam. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pp. 75–84. ACM, 2007.

Nissim, Kobbi, Raskhodnikova, Sofya, and Smith, Adam. Smooth sensitivity and sampling in private data analysis, 2011. URL http://www.cse.psu.edu/~ads22/pubs/NRS07/.

Rubinstein, Benjamin IP, Bartlett, Peter L, Huang, Ling, and Taft, Nina. Learning in a large function space: Privacy-preserving mechanisms for svm learning. *Journal of Privacy and Confidentiality*, 4(1):4, 2012.

Smith, Adam and Thakurta, Abhradeep. Nearly optimal algorithms for private online learning in full-information and bandit settings. In *NIPS*, 2013.

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

Sutton, Richard S and Barto, Andrew G. *Reinforcement learning: An introduction*. MIT press, 1998.

Szepesvári, Csaba. *Algorithms for reinforcement learning*. Morgan & Claypool Publishers, 2010.

Thakurta, Abhradeep Guha and Smith, Adam. Differentially private feature selection via stability arguments, and the robustness of the lasso. In *Conference on Learning Theory*, pp. 819–850, 2013.

Tossou, Aristide C. Y. and Dimitrakakis, Christos. Algorithms for differentially private multi-armed bandits. In *International Conference on Artificial Intelligence (AAAI 2016)*, 2016.