Journal of Machine Learning Research 18 (2017) 1–5 Submitted 7/17; Published

diffpriv: An R Package for Practical Differential Privacy

Benjamin I. P. Rubinstein BRUBINSTEIN@UNIMELB.EDU.AU
School of Computing and Information Systems
The University of Melbourne, Parkville, VIC 3010, Australia

Francesco Aldà FRANCESCO.ALDA@RUB.DE
Horst Görtz Institute for IT Security and Faculty of Mathematics
Ruhr-Universität Bochum, D-44780 Bochum, Germany

Editor: TBD

Abstract

The R package diffpriv provides tools for statistics and machine learning under differential privacy. Suitable for releasing analyses on privacy-sensitive data to (untrusted) third parties, differential privacy has become the framework of choice for privacy-preserving learning. diffpriv delivers: (a) implementations of popular generic mechanisms for privatizing non-private target functions, including the Laplace and exponential mechanisms; (b) a recent sensitivity sampler due to Rubinstein and Aldà (2017) that empirically estimates the sensitivity of non-private targets—obviating mathematical analysis for exact sensitivity bounds needed for most generic mechanisms; (c) an extensible framework for implementing differentially-private mechanism. Together, the components of diffpriv permit easy high-utility privatization of complex analyses, learners and even black-box software programs.

Keywords: differential privacy, empirical process theory, R, open-source software

1. Introduction

Differential privacy (Dwork et al., 2006) has quickly become a key framework for semantic guarantees of data privacy when releasing analysis on privacy-sensitive data to untrusted third parties. A great deal of its popularity is owed to a suite of generic mechanisms for privatizing non-private target functions of data *i.e.*, statistics, estimation procedures, and learners. Common to these generic mechanisms is the requirement that the non-private target's sensitivity to dataset perturbation is known and bounded. In all except the most trivial analyses, bounding sensitivity is prohibitively involved. This paper describes the diffpriv R package that implements generic mechanisms for differential privacy, along with our recent sensitivity sampler that replaces exact sensitivity bounds with empirical estimates (Rubinstein and Aldà, 2017). As a result, diffpriv enables most any procedure to be privatized under random differential privacy (Hall et al., 2012), without any mathematical analysis and in many cases high utility.

2. Generic Mechanisms for Differential Privacy

Fundamental to differential privacy is a privacy-sensitive *dataset* (or *database*) $D \in \mathcal{D}^n$ on *domain* $\mathcal{D}$. In diffpriv a dataset can be any `list`. We say that a pair of databases $D, D' \in \mathcal{D}^n$ is *neighboring* if they differ on exactly one record. While individual records of a $\mathcal{D}$ should not be revealed, we aim to privately release aggregate information on $\mathcal{D}$ with mechanisms. A *mechanism* is a random-valued function of databases, $M : \mathcal{D} \to \mathcal{R}$, taking values in a response set; and implemented in diffpriv as a virtual class `DPMech`. M preserves differential privacy if its response distributions are close on neighboring pairs—an adversarial observer of the mechanism, would not be able to determine a record, even with knowledge of the other $n-1$ records. For more on differential privacy, see the book (Dwork and Roth, 2014).

Definition 1 (Dwork et al., 2006) *For $\epsilon > 0$, mechanism $M : \mathcal{D}^n \to \mathcal{R}$ preserves ϵ-*differential privacy *if for all neighboring pairs $D, D' \in \mathcal{D}^n$, measurable responses $R \subseteq \mathcal{R}$, $\Pr(M(D) \in R) \leq \exp(\epsilon) \cdot \Pr(M(D') \in R)$. Alternatively for $\delta \in [0,1)$, relaxed (ϵ,δ)-*differential privacy *holds if* $\Pr(M(D) \in R) \leq \exp(\epsilon) \cdot \Pr(M(D') \in R) + \delta$.

Privacy parameters ϵ (ϵ, δ) are encapsulated in instances of diffpriv classe `DPParamsEps` (`DPParamsDel` respectively). As most mechanisms operate at tunable privacy levels, the virtual`DPMech` generic method `releaseResponse(mechanism, privacyParams, X)` takes such privacy parameters along with sensitive dataset. Most generic mechanisms in differential privacy share a number of properties leveraged by the diffpriv package as follows.

Privatizing a target function. Many mechanisms $M : \mathcal{D}^n \to \mathcal{R}$ seek to privatize a non-private *target* function $f : \mathcal{D}^n \to \mathcal{B}$, with range $\mathcal{B}$ often (but not always!) coinciding with $\mathcal{R}$. As such all objects of `DPMech` mechanisms can be initialized with a `target` slot that holds a variable of type `function`. A mechanism's `releaseResponse()` method should call `target` in forming privacy-preserving responses.

Normed target range space. Target f's output space $\mathcal{B}$ is typically imbued with a norm, denoted $\|\cdot\|_{\mathcal{B}}$, needed for measuring the sensitivity of f's outputs to input perturbation. diffpriv flexibly represents this norm within `DPMech` objects as described next.

Sensitivity-induced privacy. Many mechanisms achieve differential privacy by calibrating randomization to the sensitivity of the target function f. Targets that are relatively insensitive (sensitive) to perturbing input D to neighboring D' need relatively little (large) response randomization. On a pair of neighboring databases $D, D' \in \mathcal{D}^n$ the *sensitivity* of f is measured as $\Delta(D, D') = \|f(D) - f(D')\|_{\mathcal{B}}$. *Global sensitivity* is the largest such value $\overline{\Delta} = \sup_{D,D'} \|f(D) = f(D')\|_{\mathcal{B}}$ over all possible neighboring pairs.

As we discuss in (Rubinstein and Aldà, 2017), a broad class of generic mechanisms, taking sensitivity Δ as a parameter, are *sensitivity-induced private*: for any neighboring pair $D, D' \in \mathcal{D}^n$ if $\Delta(D, D') \leq \Delta$ then the mechanism M_Δ run with parameter Δ achieves $\Pr(M_\Delta(D) \in R) \leq \exp(\epsilon) \cdot \Pr(M_\Delta(D') \in R)$ for all $R \subseteq \mathcal{R}$. When run with $\Delta = \overline{\Delta}$, the RHS condition holds for all neighboring pairs, and $M_{\overline{\Delta}}$ satisfies ϵ-differential privacy. Similarly for (ϵ, δ)-differential privacy. Indeed this is how all proofs of differential privacy are derived, for the generic mechanisms implemented in diffpriv. `DPMech` objects can take a `sensitivity` argument at initialization stored in the slot of the same name. If the user provides a manually-derived global sensitivity bound $\overline{\Delta}$, then `releaseResponse()` responses preserve ϵ- or (ϵ, δ)-privacy (depending on the specific mechanism). This use case is demonstrated by example next.

Example: Laplace mechanism. diffpriv implements Laplace (Dwork et al., 2006) and exponential (McSherry and Talwar, 2007) mechanisms as `DPMech` subclasses `DPMechLaplace` and `DPMechExponential`. An exponential example is given in the next section. The Laplace mechanism requires that $\mathcal{B}$ be numeric $\mathbb{R}^d$ for some d, uses $\|\cdot\|_{\mathcal{B}}$ as the L_1 norm (sum of absolutes). The mechanism releases vectors in the same space $\mathcal{R} = \mathcal{B}$ by adding an i.i.d. sample of d Laplace-dstributed random variables with means 0 and scale $\overline{\Delta}/\epsilon$ to $f(D)$ to achieve ϵ-differential privacy. `DPMechLaplace` is appropriate for releasing numeric vectors.

We next demonstrate Laplace privatization of the sample mean on bounded data in $\mathcal{D}^n = [0,1]^n$, for which $\mathcal{B}$ dimension is one. Global sensitivity is readily bounded as $1/n$: For any neighboring pair $D, D' \in [0,1]^n$, $\Delta(D,D') = n^{-1}\left|\sum_{i=1}^n D_i - \sum_{i=1}^n D'_i\right|$. And since $n-1$ records are the same and the records are in $[0,1]$, this is $|D_n - D'_n|/n \leq 1/n$.

```
library(diffpriv)
f <- function(X) mean(unlist(X)) ## target function
n <- 100 ## dataset size
mechanism <- DPMechLaplace(target = f, sensitivity = 1/n, dim = 1)
D <- as.list(runif(n, min = 0, max = 1)) ## the sensitive database in [0,1]^n
pparams <- DPParamsEps(epsilon = 1) ## desired privacy budget
r <- releaseResponse(mechanism, privacyParams = pparams, X = D)
cat("Private response r$response:",   r$response,
    "\nNon-private response f(D):  ", f(D))

#> Private response r$response: 0.5167834
#> Non-private response f(D):   0.5343454
```

3. Sensitivity Sampling for Random Differential Privacy

When `target` global sensitivity is supplied as `sensitivity` within `DPMech` construction, responses are differentially private. Global sensitivity has been calculated for idealizations of *e.g.*, coefficients for regularized logistic regression (Chaudhuri and Monteleoni, 2009) and the support vector machine (Rubinstein et al., 2012; Chaudhuri et al., 2011). In complex applications such as privatizing a software function, however, `target`'s global sensitivity may not be readily available. For such cases, diffpriv implements the sensitivity sampler of Rubinstein and Aldà (2017) which forms a high-probability estimate of target sensitivity by repeated probing of sensitivity on random neighboring database pairs, leveraging tools from empirical process theory. Like sensitivity estimates, privacy holds with high probability.

Definition 2 (Hall et al., 2012) *A mechanism M preserves* (ϵ,γ)-random differential privacy *(with a corresponding form for ϵ,δ,γ) if* $\forall R \subseteq \mathcal{R}, \Pr(M(D) \in R) \leq \exp(\epsilon) \cdot \Pr(M(D') \in R)$ *holds with probability at least $1-\gamma$ over random database pairs D, D'.*

While weaker than ϵ-DP, RDP is arguably more natural than (ϵ,δ)-DP: The later safeguards all databases but not unlikely responses, while RDP protections against all responses but not pathological databases (as defined by the database sampling distribution). The sampling distribution can be anything meaningful *e.g.*, uniform, a Bayesian prior, a density from data privately fit by the Bernstein mechanism (Aldà and Rubinstein, 2017), etc.

The `DPMech` method `sensitivitySampler(object, oracle, n, m, gamma)` requires a mechanism `object`, a function `oracle` which outputs i.i.d. random databases of given size, the size of input database `n` supplied later in calls to `releaseResponse()`, a sensitivity sample size `m`, and desired privacy confidence `gamma`. Either (but not both) of `m`, `gamma` can be ommitted: the ommitted resource will be optimized automatically. For example `m` taken small (few hundred) can be used given limited time for sampling; small given `gamma` (*e.g.*, 0.05) prioritizes privacy. The sensitivity sampler calls `DPMech` method `sensitivityNorm()` which implements $\Delta(D, D')$ for the mechanism's norm and stored `target`. New sublcasses of `DPMech` need only implement this method in order to take advantage of the sensitivity sampler. Following `sensitivitySampler()`, subsequent `releaseResponse()` results have a privacy parameter slot of type `DPParamsGam` indicating response RDP.

Example: Exponential mechanism. Any sensitivity-induced `DPMech` (*e.g.*, Laplace) can be sensitivity sampled; we demonstrate the exponential mechanism here. Exponential privately optimizes an application-specific objective (or score, utility) function $s(r)$ of candidate response $r \in \mathcal{R}$, with response distribution proportional to $\exp(\epsilon \cdot s(r)/(2\Delta))$. Typically s is implicitly dependent on input D, and so `DPMechExponential` is initialized with `target` that takes D and outputs a score function. That is, $\mathcal{B} = \mathbb{R}^{\mathcal{R}}$ is a real-valued function space on $\mathcal{R}$ and the class's `sensitivityNorm()` implements the sup-norm (*cf.* Rubinstein and Aldà, 2017). In practice, users supply `target` as an R closure as demonstrated below. Given global `sensitivity` of `target`, the mechanism preserves ϵ-DP; if `sensitivitySampler()` estimates sensitivity with some `gamma`, then RDP is preserved at confidence $\gamma =$`gamma`.

Applying these ideas to find the most frequent a–z character within a dataset of top-10 computer scientists from Semantic Scholar, subject to privacy of individuals. The exponential mechanism privately maximizes total frequency. But Without bounded name lengths, this function has unbounded global sensitivity. The sensitivity sampler is therefore used for $(1, 0.1)$-RDP, with an oracle that samples representative U.S. names based on `randomNames`.

```r
library(randomNames)
oracle <- function(n) if (n==1) randomNames(1) else as.list(randomNames(n))
D <- list("Michael Jordan", "Andrew Ng", "Andrew Zisserman","Christopher Manning",
          "Jitendra Malik", "Geoffrey Hinton", "Scott Shenker",
          "Bernhard Scholkopf", "Jon Kleinberg", "Judea Pearl")
n <- length(D)
f <- function(X) { function(r) sum(r == unlist(base::strsplit(unlist(X),""))) }
rSet <- as.list(letters) ## the response set, letters a--z
mechanism <- DPMechExponential(target = f, responseSet = rSet)
mechanism <- sensitivitySampler(mechanism, oracle = oracle, n = n, gamma = 0.1)
pparams <- DPParamsEps(epsilon = 1)
r <- releaseResponse(mechanism, privacyParams = pparams, X = D)
cat("Private response r$response: ",   r$response,
    "\nNon-private f(D) maximizer:  ", letters[which.max(sapply(rSet, f(D)))])

#> Private response r$response:  e
#> Non-private f(D) maximizer:   e
```

Acknowledgments

B. Rubinstein and F. Aldà acknowledge the support of the Australian Research Council (DE160100584) and the DFG Research Training Group GRK 1817/1 respectively.

References

Francesco Aldà and Benjamin I. P. Rubinstein. The Bernstein mechanism: Function release under differential privacy. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI'2017)*, pages 1705–1711, 2017.

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

Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(Mar):1069–1109, 2011.

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

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, pages 265–284. Springer, 2006.

Rob Hall, Alessandro Rinaldo, and Larry Wasserman. Random differential privacy. *Journal of Privacy and Confidentiality*, 4(2):43–59, 2012.

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

Benjamin I. P. Rubinstein and Francesco Aldà. Pain-free random differential privacy with sensitivity sampling. In *Proceedings of the 34th International Conference on Machine Learning (ICML'2017)*, 2017. to appear; `https://arxiv.org/abs/1706.02562`.

Benjamin I. P. Rubinstein, Peter L. Bartlett, Ling Huang, and Nina Taft. Learning in a large function space: Privacy-preserving mechanisms for SVM learning. *Journal of Privacy and Confidentiality*, 4(1):65–100, 2012.