

Pain-Free Random Differential Privacy with Sensitivity Sampling

Benjamin I. P. Rubinstein
University of Melbourne, Australia

Francesco Aldà
Ruhr-Universität Bochum, Germany



RUHR
UNIVERSITÄT
BOCHUM



hgi
Horst Görtz Institut
für IT-Sicherheit

Problem & Contribution

- Generic mechanisms provide differential privacy, but require bounding global sensitivity of target.
- New sensitivity sampler instead probes target
→ Automatic + Random DP + Improved utility
- R package diffpriv released on CRAN, GitHub.

Bounding Sensitivity for Generic Mechanisms

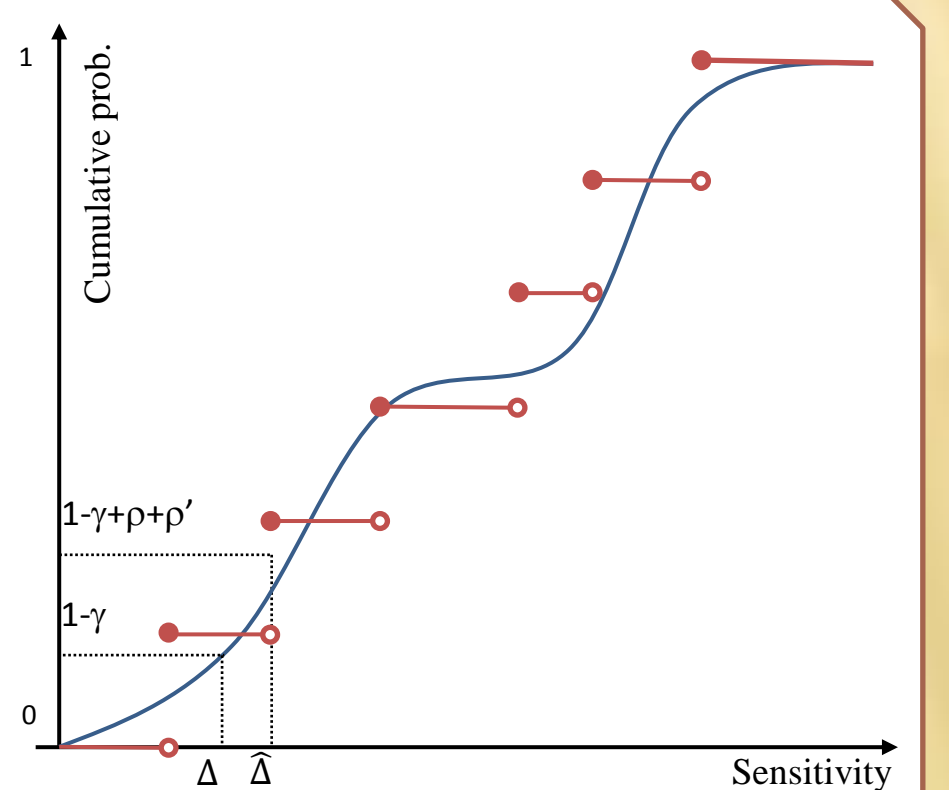
- Privatise a target function f maps DB D into B
- Sensitivity $\Delta f(D, D') = \|f(D) - f(D')\|_B$
- DP: calibrate randomisation to $\bar{\Delta}f = \sup_{D \cong D'} \Delta f(D, D')$
- Laplace mechanism: add zero-mean $\text{Lap}(\bar{\Delta}f/\epsilon)$ to f
- Others: Gaussian, exponential, Bernstein mechanisms

Prohibitive
math analysis

Algorithm: Sensitivity Sampler

Input: Target f , DB size n , distribution P , sample size m , order statistic index k ,

- Repeat $i = 1 \dots m$
 - Sample $D \cong D'$ from P
 - $G_i = \Delta f(D, D') = \|f(D) - f(D')\|_B$
- Return estimated sensitivity $\Delta f = G_{(k)}$



Privacy: Who Cares?

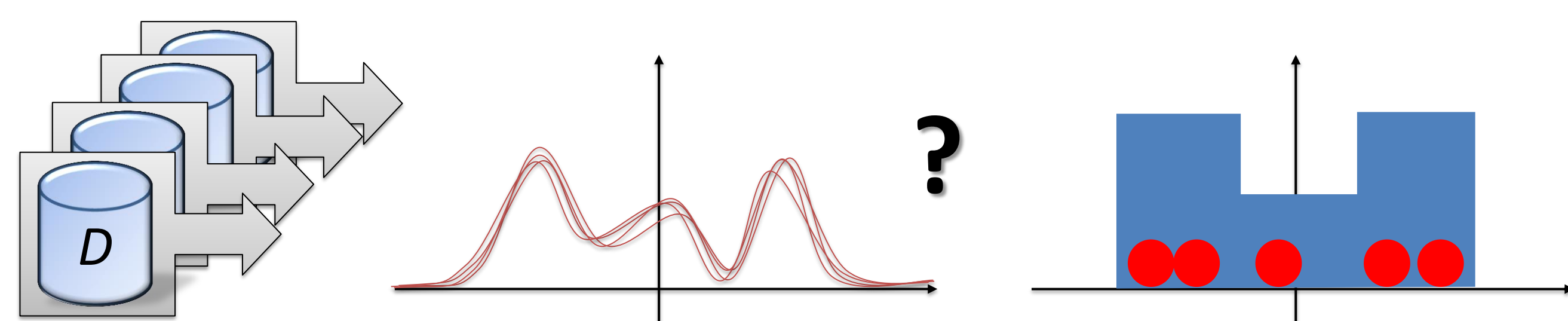


- DP deployed: Google, Apple, Uber, etc.
- Active groups: Harvard, Berkeley, CMU, Weizman, Oxford, UCSD, Stanford, etc.
- 2017 Gödel Prize to Dwork et al.

Differential Privacy [Dwork et al. 2006]

Release aggregate info, protect individual data.

- Database D : a sequence of n records, some domain
- Neighbouring DBs $D \cong D'$: differ on one record
- Mechanism M : maps DB D to random response $R \in B$ some normed space
- M has ϵ -differential privacy if, $\forall D \cong D', \forall R \subseteq B$
 $\Pr(M(D) \in R) \leq \exp(\epsilon) \cdot \Pr(M(D') \in R)$ where $\epsilon > 0$



Response indistinguishable on changing one record

- Semantic guarantee: Limits what adversary can do with: unbounded computation; knowledge of DB up to a record; knowledge of M up to randomness.

Random Differential Privacy [Hall et al. 12]

M has (ϵ, γ) -RDP for $\epsilon > 0, \gamma \in (0,1)$ if whp $1 - \gamma$ over $D \cong D': \forall R \subseteq B, \Pr(M(D) \in R) \leq \exp(\epsilon) \cdot \Pr(M(D') \in R)$

- DP: indistinguishable responses over all DB pairs
- RDP: indistinguishability over all—but *pathological*-DBs

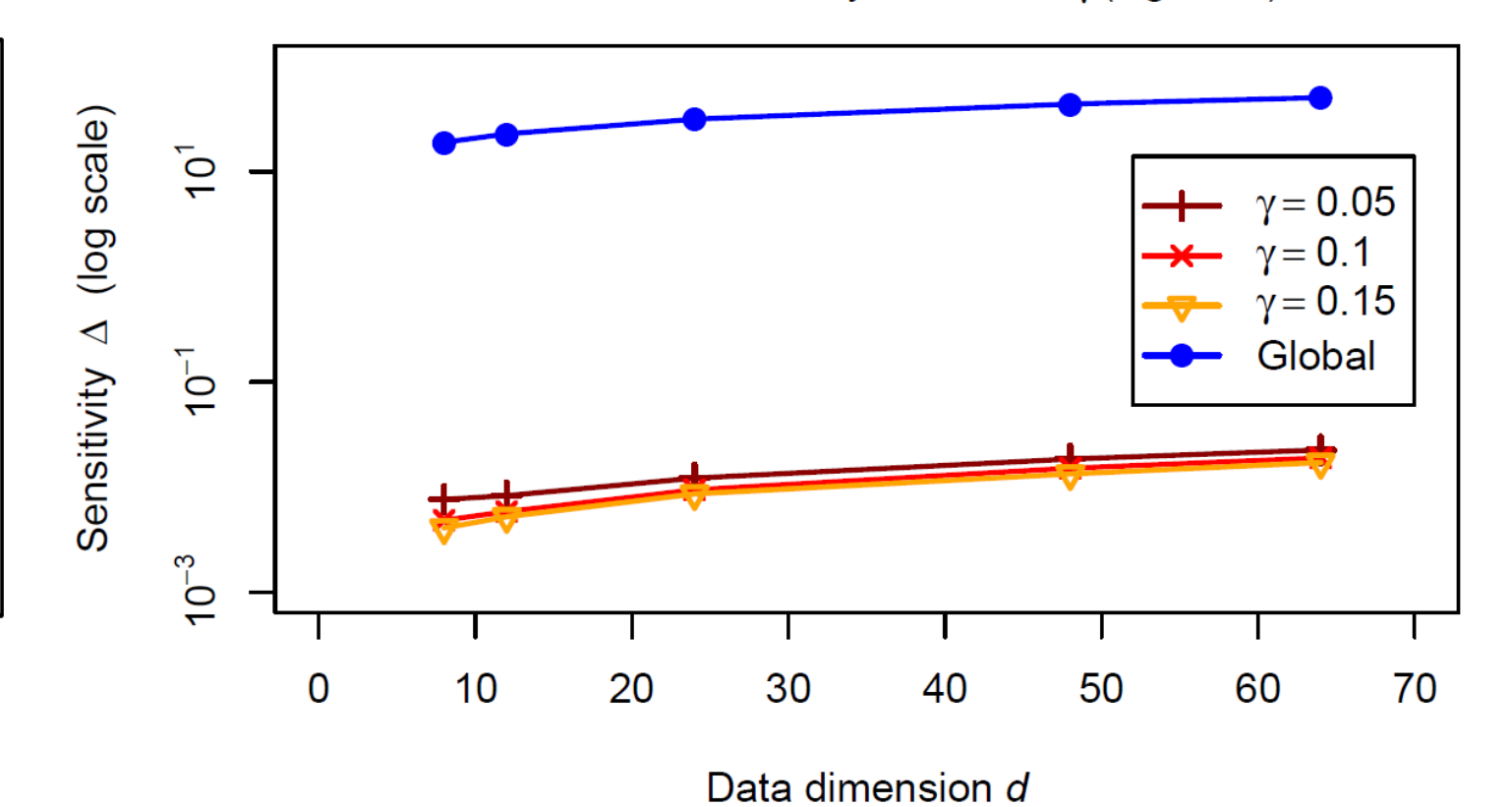
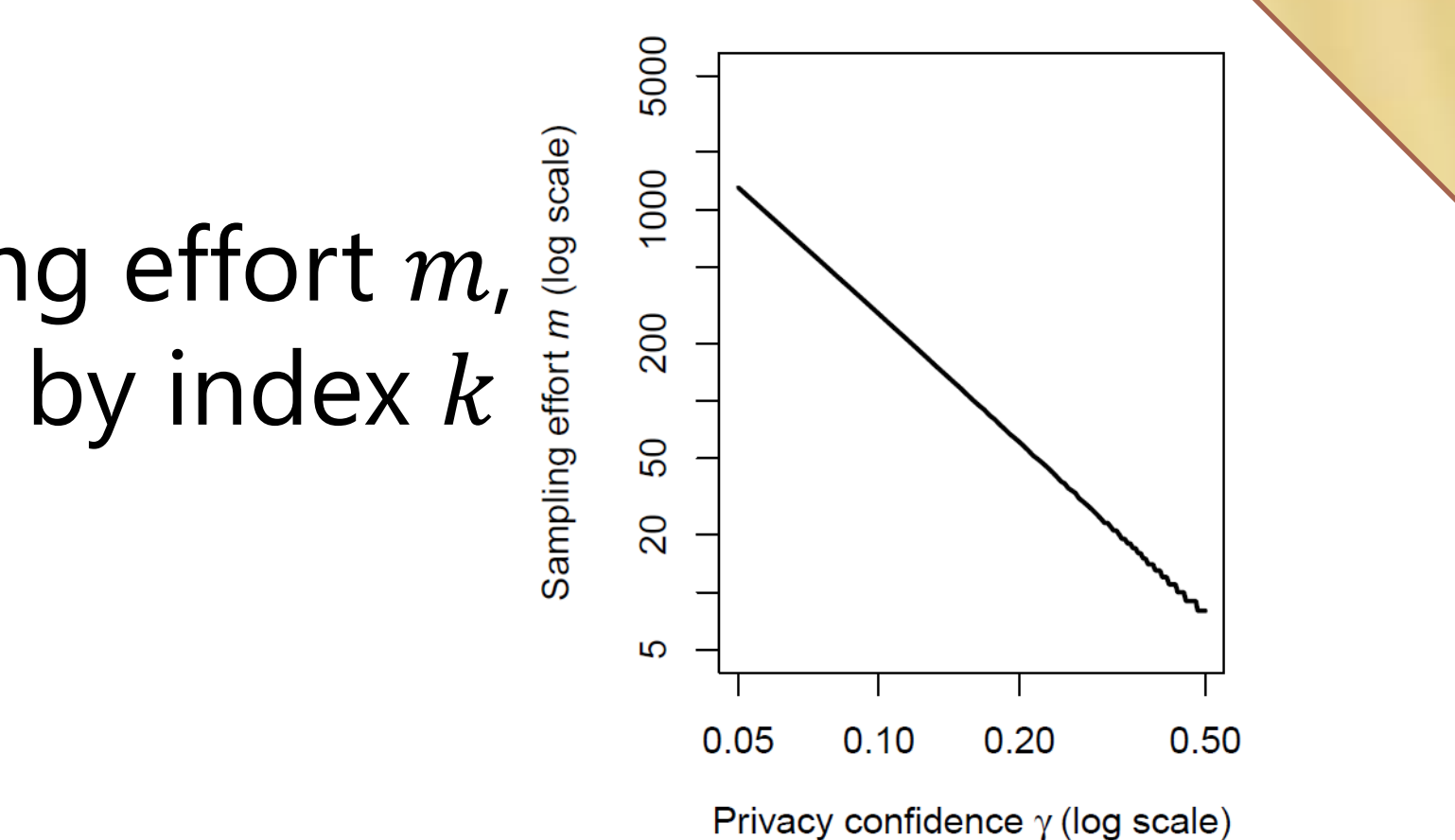
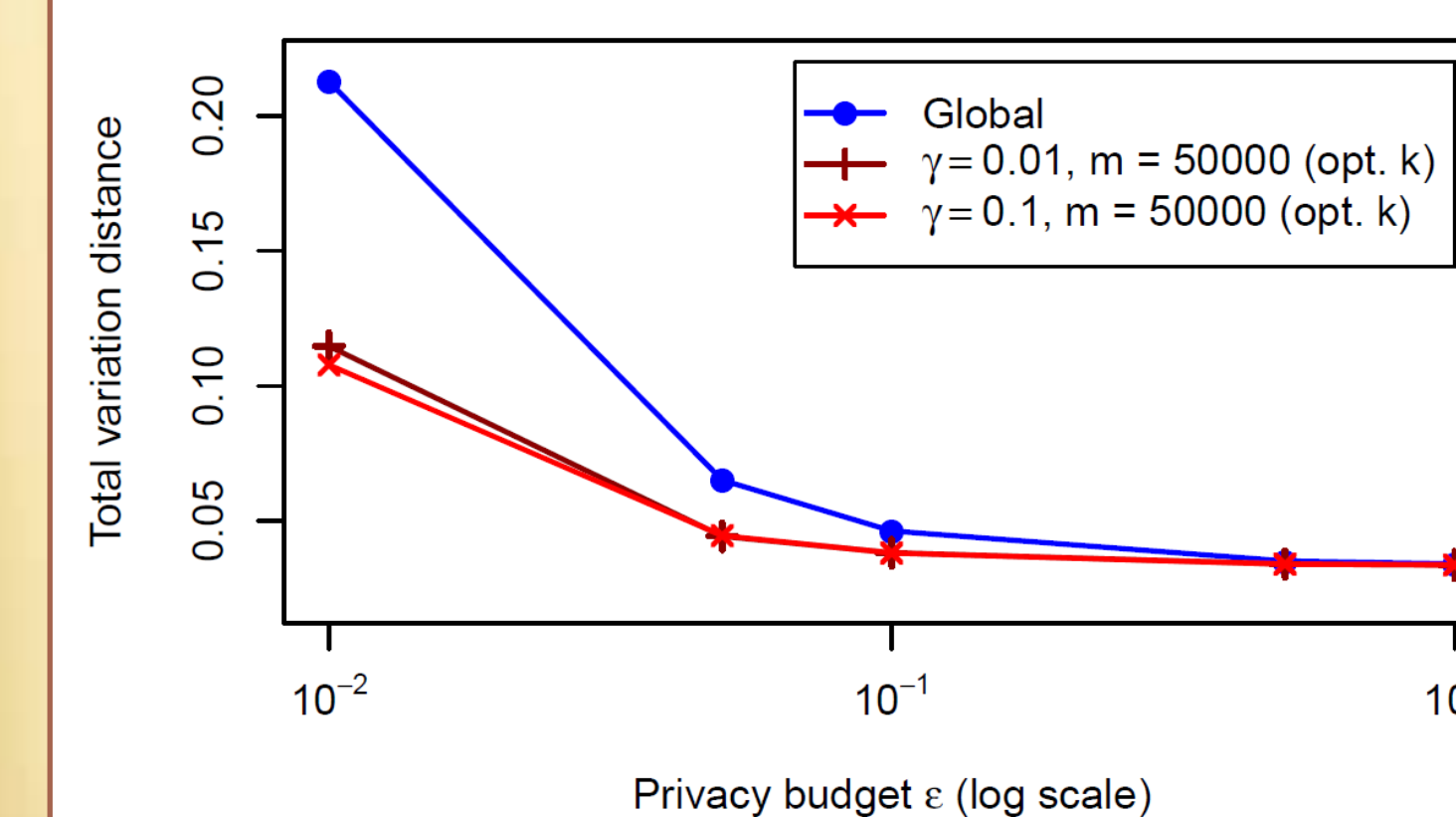
Sensitivity-Induced Private M : if for $D \cong D'$, $\Delta f(D, D') \leq \Delta$ implies ϵ -DP holds for M_Δ on pair D, D'

- $\Pr(\Delta f(D, D') \leq \Delta) \geq 1 - \gamma$ implies M_Δ has (ϵ, γ) -RDP
- Given CDF of $\Delta f(D, D')$ get Δ by inverting CDF at $1 - \gamma$
- Empirical CDF of iid sample estimates CDF uniformly

Results

Can optimise any of: sampling effort m , RDP confidence γ , utility by index k

Utility vs Privacy for kernel density estimation



SVM sensitivity: global vs sampled

Open-Source Package diffpriv

diffpriv

Overview

The diffpriv package makes privacy-aware data science in R easy. diffpriv implements the formal framework of differential privacy: differentially private mechanisms can safely release to untrusted third parties statistics computed, models fit, or arbitrary structures derived on privacy-sensitive data. Due to the worst-case nature of the framework, mechanism development typically requires involved theoretical analysis. diffpriv offers a turn-key approach to differential privacy by automating this process with sensitivity sampling in place of theoretical sensitivity analysis.

Installation

Obtaining diffpriv is easy. From within R:

```
## Install the release version of diffpriv from CRAN:
install.packages("diffpriv")

## Install the latest development version of diffpriv from GitHub:
install.packages("devtools")
devtools::install_github("brubinstein/diffpriv")
```

Example

A typical example in differential privacy is privately releasing a simple target function of privacy-sensitive input data x . Say the mean of numeric data:

```
## as a target function we'd like to run on private data x, releasing the result
target <- function(x) mean(x)
```

First load the diffpriv package (installed as above) and construct a chosen differentially-private mechanism for privatizing target.

```
## target seeks to release a numeric, so we'll use the Laplace mechanism-->
```

diffpriv

Links

Download from CRAN at: <https://cran.r-project.org/package=diffpriv>
Browse source code at: <https://github.com/brubinstein/diffpriv>
Report a bug at: <https://github.com/brubinstein/diffpriv/issues>

License

MIT + Rie LICENSE

Citation

Citing diffpriv

Developers

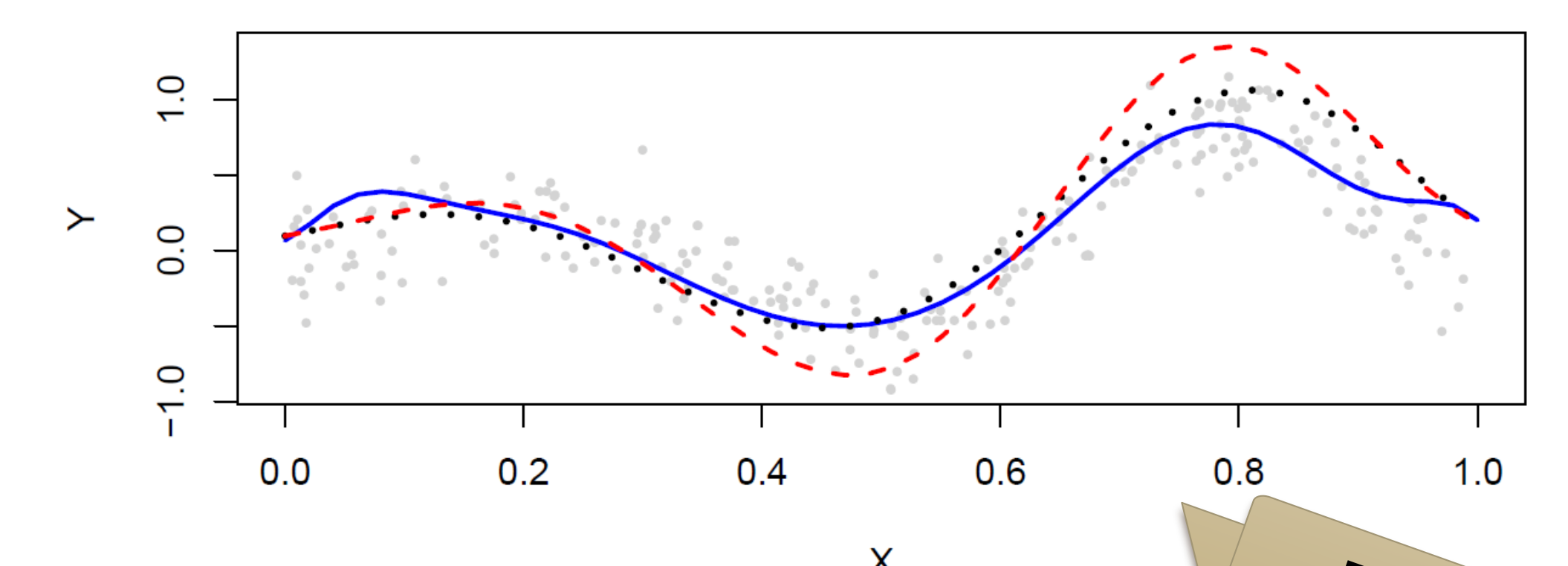
Benjamin Rubinstein
Author, maintainer
Francesco Aldà
Author

Dev status

CRAN version: 0.2.0 (2020-08-10)
CRAN date: 2020-08-10
CRAN status: OK
CRAN type: R
CRAN file: diffpriv_0.2.0.tar.gz

- Open-source R package on GitHub
- 'Official' on CRAN with rigorous submission process checks
- Implements generic mechanisms for DP, sampler for automatic RDP
- roxygen2 docs
- Tutorial vignettes (→ JMLR MLOSS)
- 98% code coverage with testthat
- Travis CI continuous integration

Priestly-Chao Kernel Regression



`install.packages("diffpriv")`

Easy
install

```
library(diffpriv)
m <- DPMechBernstein(
  target=pck_regression, latticeK=K, dims=1)
m <- sensitivitySampler(m, oracle=P, gamma=0.05)
R <- releaseResponse(m, DPPParams(epsilon=1), D)
```