CS208: Applied Privacy for Data Science
Machine Learning under DP

School of Engineering & Applied Sciences
Harvard University

March 1, 2022

Following slides from:

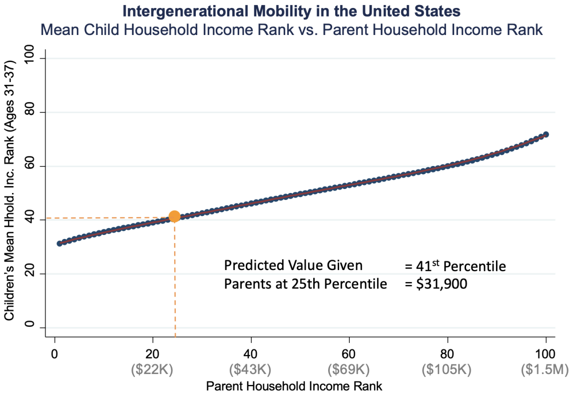
Practical Method to Reduce Privacy Loss when Disclosing Statistics Based on Small Samples

Raj Chetty, Harvard University and NBER
John N. Friedman, Brown University and NBER

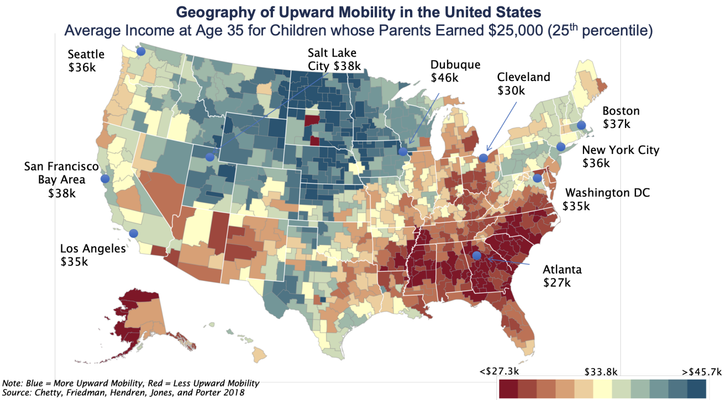
March 2019

Publishing Statistics Based on Small Cells

- Social scientists increasingly use confidential data to publish statistics based on cells with a small number of observations
- Causal effects of schools or hospitals [e.g., Angrist et al. 2013, Hull 2018]
- Local area statistics on health outcomes or income mobility [e.g., Cooper et al. 2015, Chetty et al. 2018]



Source: Chetty, Friedman, Hendren, Jones, Porter (2018)



Geography of Upward Mobility in the United States
Average Income at Age 35 for Children whose Parents Earned $25,000 (25th percentile)
Seattle $36k
Salt Lake City $38k
Dubuque $46k
Cleveland $30k
Boston $37k
New York City $36k
San Francisco Bay Area $38k
Washington DC $35k
Los Angeles $35k
Atlanta $27k
<$27.3k
$33.8k
>$45.7k
Note: Blue = More Upward Mobility, Red = Less Upward Mobility
Source: Chetty, Friedman, Hendren, Jones, and Porter 2018

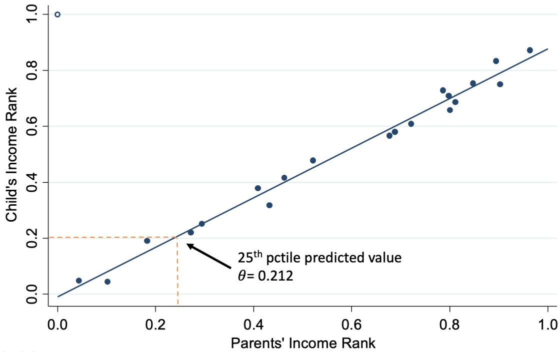
Controlling Privacy Loss

- Problem with releasing such estimates at smaller geographies (e.g., Census tract): risk of disclosing an individual's data
- Literature on differential privacy has developed practical methods to protect privacy for simple statistics such as means and counts [Dwork 2006, Dwork et al. 2006]
- But methods for disclosing more complex estimates, e.g. regression or quasiexperimental estimates, are not feasible for many social science applications [Dwork and Lei 2009, Smith 2011, Kifer et al. 2012]

This Paper: A Practical Method to Reduce Privacy Loss

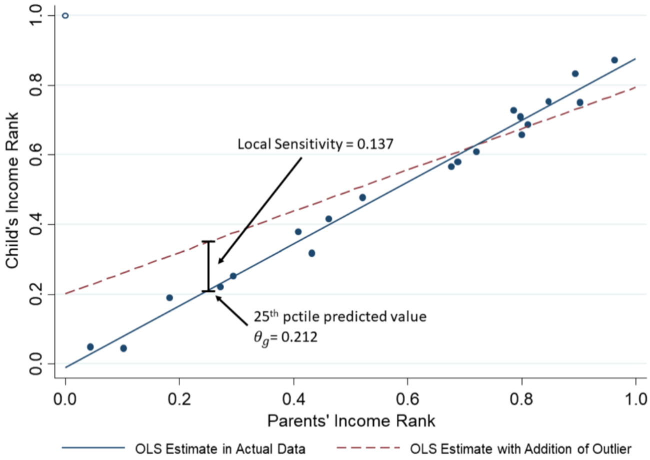
- We develop and implement a simple method of controlling privacy loss when disclosing arbitrarily complex statistics in small samples
  - The “Maximum Observed Sensitivity” (MOS) algorithm
- Method outperforms widely used methods such as cell suppression both in terms of privacy loss and statistical accuracy
  - Does not offer a formal guarantee of privacy, but potential risks occur only at more aggregated levels (e.g., the state level)

Example Regression from One Small Cell



Source: Authors' simulations.

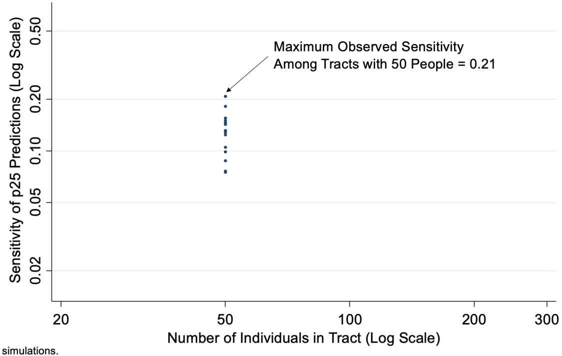
Figure 1: Calculation of local sensitivity



Maximum Observed Sensitivity

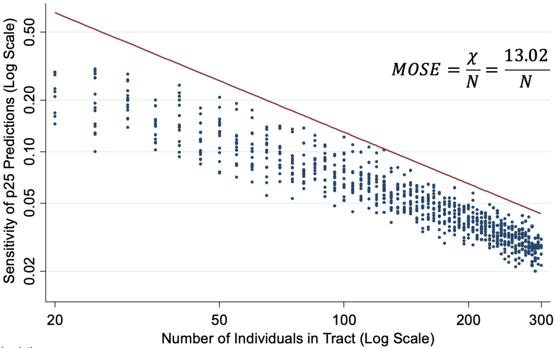
- Our method: use the maximum observed local sensitivity across all cells in the data
  - In geography of opportunity application, calculate local sensitivity in every tract
  - Then use the maximum observed sensitivity (MOS) across all tracts within a given state as the sensitivity parameter for every tract in that state
- Analogous to Empirical Bayes approach of using actual data to construct prior on possible realizations rather than considering all possible priors

Maximum Observed Sensitivity Envelope



Source: Authors' simulations.

Computing Maximum Observed Sensitivity



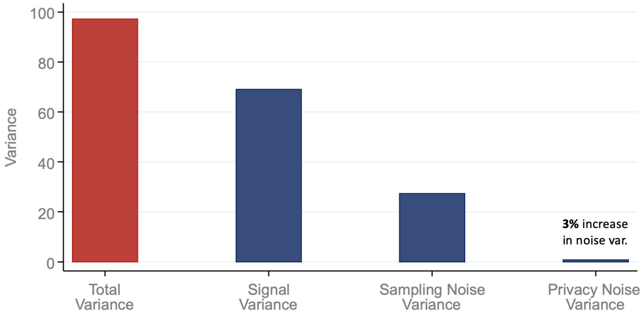
Source: Authors' simulations.

Producing Noise-Infused Estimates for Public Release

- Main lesson: tools from differential privacy literature can be adapted to control privacy loss while improving statistical inference
  - Opportunity Atlas has been used by half a million people, by housing authorities to help families move to better neighborhoods, and in downstream research [Creating Moves to Opportunity Project; Morris et al. 2018]
  - The MOS algorithm can be practically applied to any empirical estimate
- Example: difference-in-differences or regression discontinuity
  - Even when there is only one quasi-experiment, pretend that a similar change occurred in other cells of the data and compute MOS across all cells

Variance Decomposition for Tract-Level Estimates

Teenage Birth Rate For Black Women With Parents at 25th Percentile



Source: Chetty, Friedman, Hendren, Jones, Porter (2018)

Conclusion

- Use max observed sensitivity χ, tract counts, and exogenously specified privacy parameter ϵ to add noise and construct public estimates:

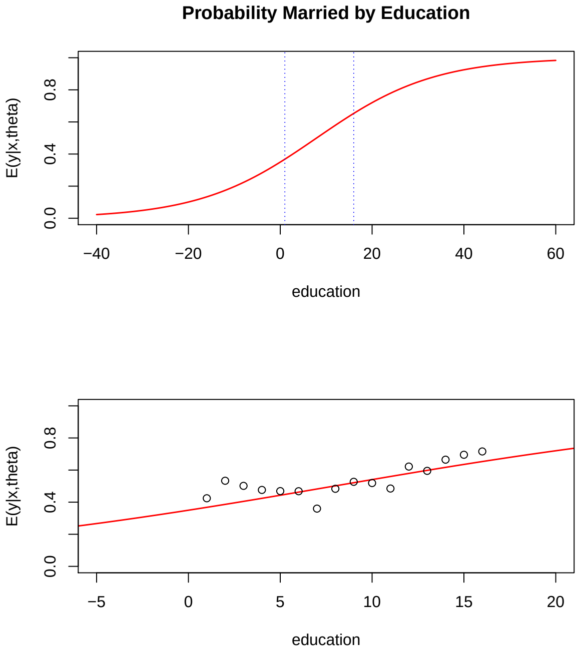
$$\tilde{\theta}_g = \theta_g + L\left(0, \frac{\chi}{\epsilon N_g}\right) \quad \tilde{N}_g = N_g + L\left(0, \frac{1}{\epsilon}\right)$$

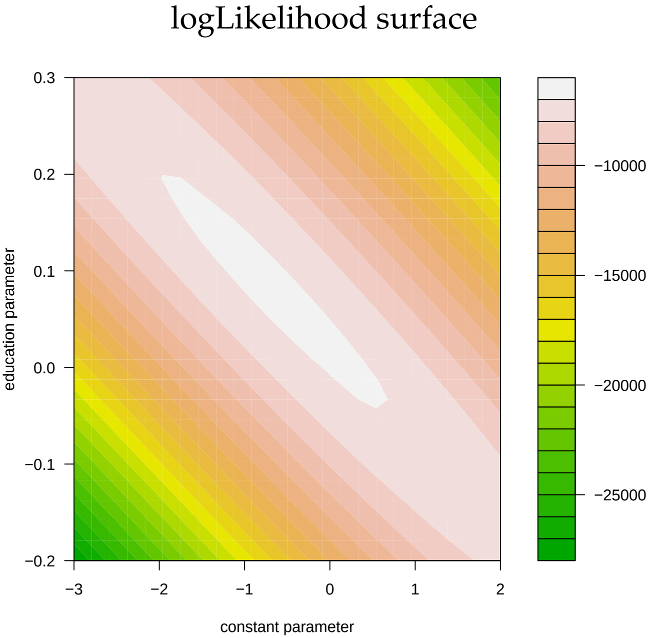
  - This method not "provably private," but it reduces privacy risk to release of the single max observed sensitivity parameter (!)
  - Privacy loss from release of regression statistics themselves is controlled below risk tolerance threshold ϵ.
- Critically, χ can be computed at a sufficiently aggregated level that disclosure risks are considered minimal ex-ante

DP Optimization of Complex Models

Logit Model

$$
\begin{aligned}
logL(y|x,\theta) &= \sum_{i=1}^{N} y_i \log(\pi_i) + (1-y_i)\log(1-\pi_i), \\
\pi_i &= \frac{1}{1+e^{-\beta_0-\beta_1 x_i}}.
\end{aligned}
$$


Probability Married by Education
E(y|x,theta)
0.0
0.4
0.8
−40
−20
0
20
40
60
education
E(y|x,theta)
0.0
0.4
0.8
−5
0
5
10
15
20
education


logLikelihood surface
education parameter
constant parameter
0.3
0.2
0.1
0.0
−0.1
−0.2
−3
−2
−1
0
1
2
−10000
−15000
−20000
−25000

Algorithm 1 Differentially private SGD (Outline)

Input: Examples $\{x_1, \ldots, x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i)$. Parameters: learning rate η_t, noise scale σ, group size L, gradient norm bound C.

Initialize θ_0 randomly

for $t \in [T]$ **do**

Take a random sample L_t with sampling probability L/N

Compute gradient

For each $i \in L_t$, compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t}\mathcal{L}(\theta_t, x_i)$

Clip gradient

$\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i)/\max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)$

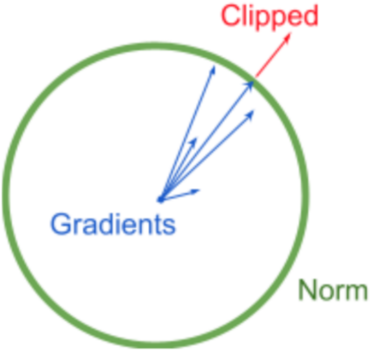
Add noise

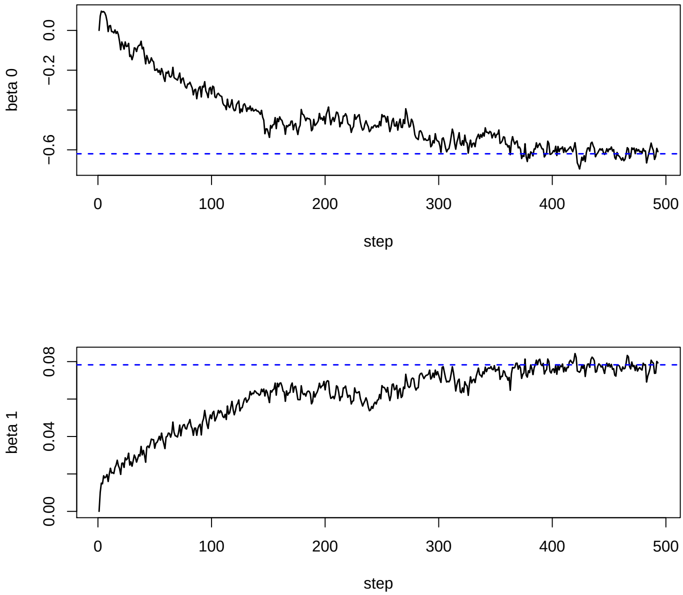
$\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}\left(\sum_i \bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$

Descent

$\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$

Output θ_T and compute the overall privacy cost (ε, δ) using a privacy accounting method.


Clipped
Gradients
Norm


beta 0
0.0
−0.2
−0.6
0
100
200
300
400
500
step
beta 1
0.00
0.04
0.08
0
100
200
300
400
500
step