Differentially Private Data Release under Partial Information

David Zeber Martin Lopatka

Systems Research Group, Mozilla

PEPR, 12 August 2019

Data and privacy

User data!

Utility:

- Learning about users
  - build insights or models by computing summaries & aggregations
- Sharing
  - internally across different parts of the org
  - with business partners
  - publicly

What are the privacy implications for sensitive data?

Differential Privacy as a solution

Differential Privacy (DP) is the state-of-the-art for releasing reports/views based on sensitive data

- avoids revealing “too much” about the users in the dataset

E.g. a DP view allows for further analysis without needing access to the raw data

- can be shared with a partner

Differential Privacy

- A specific guarantee to individuals whose data is included in a dataset
- Limits how much can be learnt about an individual from a DP-protected dataset
- Limit is quantified by a number (ϵ): **privacy budget**

DP mechanisms

To use DP, apply a randomized operation (**mechanism**) to the dataset.

- Original data is masked by injecting randomness in a strategic way
- Makes “influential” values in the dataset less influential

Applying DP

In practice, DP is non-trivial to apply

- Real-world datasets are more complex than what is typically described in the literature
- Multiple mechanisms may be available

Selecting a “good” mechanism is important:

- a bad choice could end up giving low utility results while consuming privacy budget

Selecting a DP mechanism

- Which mechanism will perform best depends on the properties of the particular dataset
- "Peeking" at the data violates DP

Solutions:

- draw on literature set in the same domain
- allocate some of the privacy budget to learning characteristics in a DP way

Existing partial information

In practice, we often do have prior partial knowledge about the dataset:

- may satisfy known constraints
  - e.g. limitations of an app or reasonable user behaviour
- other summary information about the users may already have been shared
  - e.g. `addons.mozilla.org` lists raw installation counts

Partial information example

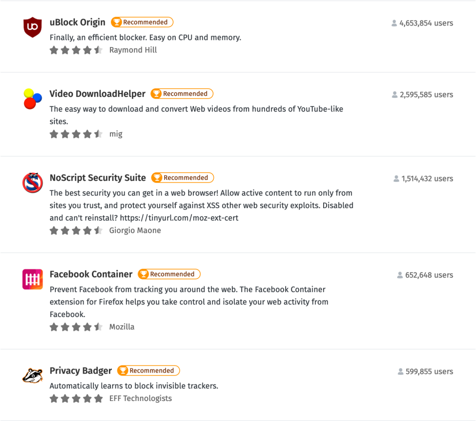


Figure: `addons.mozilla.org` (recommended add-ons)

Proposed DP selection approach

Data model: describe a dataset probabilistically in terms of how it was produced

- i.e. as a generative distribution over possible datasets
- known constraints built in via conditioning

Mechanism selection: evaluate candidate DP mechanisms by simulating over a sample of possible datasets

Proposed DP selection approach

Benefits:

- Can incorporate partial information of differing specificities
- Does not consume any privacy budget
- Does not require access to the private dataset (e.g. could be performed by a third party)

Problem setting

Data model: user-item data

- $n \times m$ user-item matrix
- entry $(i, j) = 1$ if user i interacted with item j

$$
\begin{array}{cc}
 & \text{(}\textit{items}\text{)} \\
 & \begin{array}{cccc} \mathrm{A} & \mathrm{B} & \mathrm{C} & \mathrm{D} \end{array} \\
\textit{(users)} \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}
\end{array}
$$

User-item data: output

Goal: report the item frequency counts with privacy protection

- labeled column sums of the matrix

$$\begin{array}{cc} & \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \end{array} \xrightarrow{\text{colsums}} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{bmatrix} 4 & 2 & 2 & 1 \end{bmatrix} \end{array}$$

Influence of a single user: adds 1 to the count for each item they report

User-item data: mechanism

1. Limit each user to k items selected at random
2. Compute item frequency counts over the limited items
3. Add $Lap(k/\epsilon)$ random noise to each count

Satisfies ϵ-DP

User-item data: mechanism

$$k = 1$$

$$\begin{array}{c c} & \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \end{array} \xrightarrow{\text{limit}} \begin{array}{c c} & \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 1 & 0 & \color{red}{0} & \color{red}{0} \\ 0 & 1 & 0 & 0 \\ \color{red}{0} & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & \color{red}{0} & 0 & 0 \end{bmatrix} \end{array}$$

$$\xrightarrow{\text{colsums}} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{bmatrix} 3 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

$$\xrightarrow{\text{noise}} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{bmatrix} 2.52 & 2.64 & 0.82 & 1.17 \end{bmatrix} \end{array}$$

User-item mechanism tradeoff

This introduces a bias-variance tradeoff tuned by the parameter k

- If many users have close to m items, would want k to be large (lower bias)
- If most users have few items, would want k to be small (lower noise)

Case study

- A team at Mozilla curates a dataset containing a list of “favourite” sites (URLs) from a subset of Firefox users*
- Access to raw data is highly restricted for privacy reasons
- Data curators are willing to release a DP version of the frequency distribution if they are provided a query to run

* who have opted in to this data collection

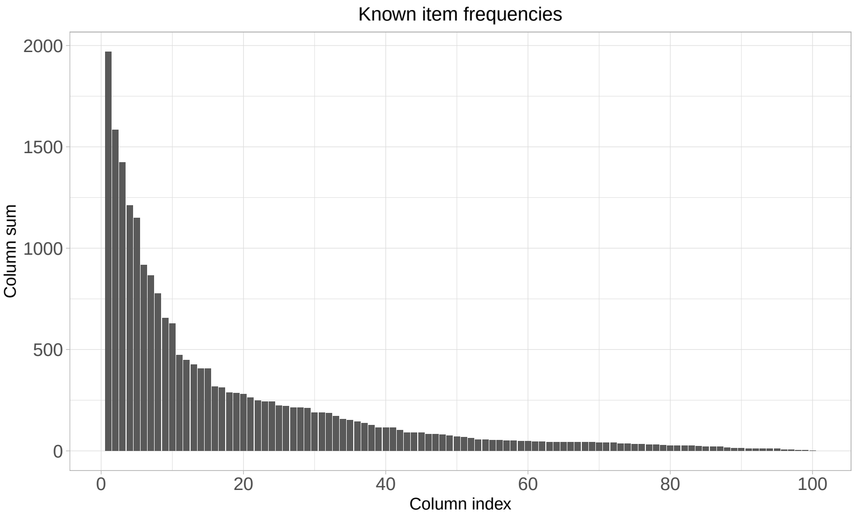
Case study

What do we know about this data?

- Under the policies governing this dataset, the curators are able to share internally the site counts (i.e. frequency distribution) *with URLs scrubbbed*

We can use this to build a generative model for this dataset with constraints.

Item frequencies



Generative model: setting

- Dataset is an n (users) $\times$ m (items) binary matrix
- Fix column ordering from highest to lowest column sum, same for rows
- Known: m, column sums $n \geq c_1 \geq \cdots \geq c_m \geq 1$

Generative model: number of rows

In this case, number of users n is not known directly. However, column sums impose constraints:

- $n \geq c_1 = \max c_j$ (single copy of each item per user)
- $n \leq \sum_j c_j$ (at least 1 item per user)

Examples: number of rows

$$\begin{array}{cccc} A & B & C & D \end{array}$$
$$\begin{bmatrix} 4 & 2 & 2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$n = 4 \qquad n = 5 \qquad n = 8$$

Generative model: row sums

Given n, we have constraints on the row sums (number of items reported by each user):

- $m \geq r_1 \geq \cdots \geq r_n \geq 1$
- $\sum_{i=1}^{n} \min(r_i, p) \geq \sum_{j=1}^{p} c_j$ for $p = 1, \ldots, m$ (Gale-Ryser conditions for binary matrices)

These can be reformulated into sequential bounds $L_i \leq r_i \leq U_i$ where $L_i = L(r_1, \ldots, r_{i-1})$ and $U_i = U(r_1, \ldots, r_{i-1})$

Examples: row sums

$$\begin{array}{cccc} A & B & C & D \\ \end{array}$$
$$\begin{bmatrix} 4 & 2 & 2 & 1 \end{bmatrix}, \quad n = 5$$

$$\begin{array}{c} 4 \\ 2 \\ 1 \\ 1 \\ 1 \end{array} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad \begin{array}{c} 3 \\ 2 \\ 2 \\ 1 \\ 1 \end{array} \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \qquad \begin{array}{c} 2 \\ 2 \\ 2 \\ 2 \\ 1 \end{array} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Examples: matrices

$$\begin{array}{cccc} A & B & C & D \\ \end{array}$$

$$\begin{bmatrix} 4 & 2 & 2 & 1 \end{bmatrix}, \quad n = 5, \quad \boldsymbol{r} = [3, 2, 2, 1, 1]$$

$$\begin{array}{c} 3 \\ 2 \\ 2 \\ 1 \\ 1 \end{array} \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \begin{array}{c} 3 \\ 2 \\ 2 \\ 1 \\ 1 \end{array} \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \qquad \begin{array}{c} 3 \\ 2 \\ 2 \\ 1 \\ 1 \end{array} \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Generative model

1. Select n uniformly among possible choices
2. Given n, select $\boldsymbol{r}$ uniformly subject to constraints
3. Given marginal counts, select a binary matrix uniformly

Performed using MCMC sampling (cf. Verhelst, 2007)

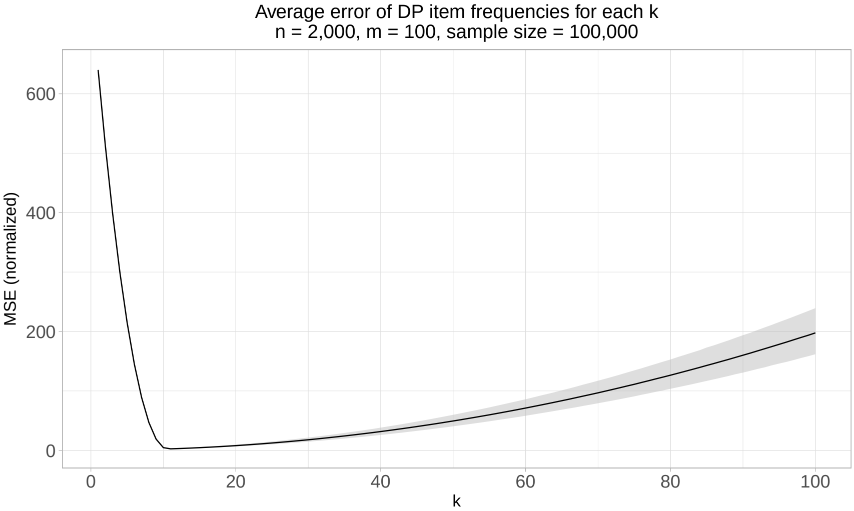
Mechanism selection

Given a feasible matrix:

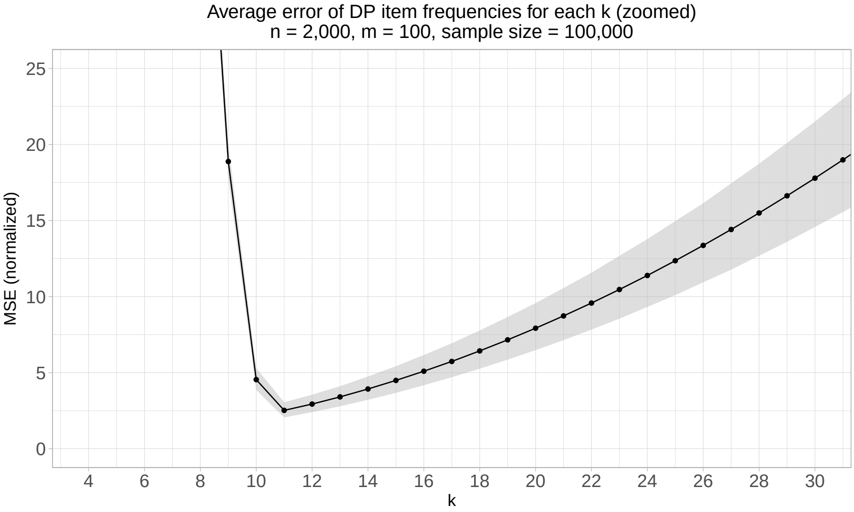
1. Simulate the DP mechanism for different values of k with the desired ϵ
2. Compute squared-error loss relative to the true item counts
3. Estimate MSE by averaging over multiple replicates of (1) and (2)

Replicate the above over a sample of feasible matrices

Evaluation results



Evaluation results



Conclusion

We present an alternative approach to mechanism selection for DP:

- does not consume privacy budget
- does not require access to the private dataset
- leverages any available ancillary information about the problem setting

Conclusion

Vision: a software package applicable to a number of domains assisting in

- formulating constraints
- running simulations
- visualizing and interpreting results

Thank you!