Locally Private Release of Marginal Statistics

Graham Cormode

g.cormode@warwick.ac.uk

Tejas Kulkarni (Warwick)

Divesh Srivastava (AT&T)

THE UNIVERSITY OF WARWICK

Privacy with a coin toss

Perhaps the simplest possible formal privacy algorithm:

- Scenario. Each user has a single private bit of information
  - Encoding e.g. political/sexual/religious preference, illness, etc.

THE UNIVERSITY OF WARWICK

Privacy with a coin toss

Perhaps the simplest possible formal privacy algorithm:

- **Scenario.** Each user has a single private bit of information
  - Encoding e.g. political/sexual/religious preference, illness, etc.
- **Algorithm.** Toss a (biased) coin, and
  - With probability $p > ½$, report the true answer
  - With probability $1-p$, lie

THE UNIVERSITY OF WARWICK

Privacy with a coin toss

Perhaps the simplest possible formal privacy algorithm:

- **Scenario.** Each user has a single private bit of information
  - Encoding e.g. political/sexual/religious preference, illness, etc.
- **Algorithm.** Toss a (biased) coin, and
  - With probability $p > ½$, report the true answer
  - With probability $1-p$, lie
- **Aggregation.** Collect responses from a large number N of users
  - Can 'unbias' the estimate (if we know p) of the population fraction
  - The error in the estimate is proportional to $1/\sqrt{N}$

Privacy with a coin toss

Perhaps the simplest possible formal privacy algorithm:

- **Scenario.** Each user has a single private bit of information
  - Encoding e.g. political/sexual/religious preference, illness, etc.
- **Algorithm.** Toss a (biased) coin, and
  - With probability $p > ½$, report the true answer
  - With probability $1-p$, lie
- **Aggregation.** Collect responses from a large number N of users
  - Can ‘unbias’ the estimate (if we know p) of the population fraction
  - The error in the estimate is proportional to $1/\sqrt{N}$
- **Analysis.** Gives differential privacy with parameter $\varepsilon = \ln (p/(1-p))$
  - Works well in theory, but would anyone ever use this?

Privacy in practice

Privacy in practice

- Differential privacy based on coin tossing is widely deployed
  - In Google Chrome browser, to collect browsing statistics
  - In Apple iOS and MacOS, to collect typing statistics
  - This yields deployments of over 100 million users

Privacy in practice

- Differential privacy based on coin tossing is widely deployed
  - In Google Chrome browser, to collect browsing statistics
  - In Apple iOS and MacOS, to collect typing statistics
  - This yields deployments of over 100 million users
- The model where users apply differential privately and then aggregated is known as "Local Differential Privacy"
  - The alternative is to give data to a third party to aggregate
  - The coin tossing method is known as 'randomized response'

Privacy in practice

- Differential privacy based on coin tossing is widely deployed
  - In Google Chrome browser, to collect browsing statistics
  - In Apple iOS and MacOS, to collect typing statistics
  - This yields deployments of over 100 million users
- The model where users apply differential privately and then aggregated is known as "Local Differential Privacy"
  - The alternative is to give data to a third party to aggregate
  - The coin tossing method is known as 'randomized response'
- Local Differential privacy is state of the art in 2017: Randomized response invented in 1965: five decade lead time!

THE UNIVERSITY OF WARWICK

Going beyond 1 bit of data

1 bit can tell you a lot, but can we do more?

- Recent work: materializing marginal distributions
  - Each user has d bits of data (encoding sensitive data)
  - We are interested in the distribution of combinations of attributes

Going beyond 1 bit of data

1 bit can tell you a lot, but can we do more?

- Recent work: materializing marginal distributions
  - Each user has d bits of data (encoding sensitive data)
  - We are interested in the distribution of combinations of attributes

	Gender	Obese	High BP	Smoke	Disease
Alice	1	0	0	1	0
Bob	0	1	0	1	1
…					
Zayn	0	0	1	0	0

THE UNIVERSITY OF WARWICK

Going beyond 1 bit of data

1 bit can tell you a lot, but can we do more?

- Recent work: materializing marginal distributions
  - Each user has d bits of data (encoding sensitive data)
  - We are interested in the distribution of combinations of attributes

	Gender	Obese	High BP	Smoke	Disease
Alice	1	0	0	1	0
Bob	0	1	0	1	1
…					
Zayn	0	0	1	0	0

Gender/Obese	0	1
0	0.28	0.22
1	0.29	0.21

Disease/Smoke	0	1
0	0.55	0.15
1	0.10	0.20

Nail, meet hammer

- Could apply Randomized Reponse to each entry of each marginal
  - To give an overall guarantee of privacy, need to change p
  - The more bits released by a user, the closer p gets to ½ (noise)

Nail, meet hammer

- Could apply Randomized Reponse to each entry of each marginal
  - To give an overall guarantee of privacy, need to change p
  - The more bits released by a user, the closer p gets to ½ (noise)
- Need to design algorithms that minimize information per user

THE UNIVERSITY OF WARWICK

Nail, meet hammer

- Could apply Randomized Reponse to each entry of each marginal
  - To give an overall guarantee of privacy, need to change p
  - The more bits released by a user, the closer p gets to ½ (noise)
- Need to design algorithms that minimize information per user
- First observation: a sampling trick
  - If we release n bits of information per user, the error is $n/\sqrt{N}$
  - If we sample 1 out of n bits, the error is $\sqrt{(n/N)}$
  - Quadratically better to sample than to share!

THE UNIVERSITY OF WARWICK

What to materialize?

Different approaches based on how information is revealed

What to materialize?

Different approaches based on how information is revealed

1. We could reveal information about all marginals of size k
   - There are (d choose k) such marginals, of size 2^k each

What to materialize?

Different approaches based on how information is revealed

1. We could reveal information about all marginals of size k
   - There are (d choose k) such marginals, of size 2^k each
2. Or we could reveal information about the full distribution
   - There are 2^d entries in the d-dimensional distribution
   - Then aggregate results here (obtaining additional error)

What to materialize?

Different approaches based on how information is revealed

1. We could reveal information about all marginals of size k
   - There are (d choose k) such marginals, of size 2^k each
2. Or we could reveal information about the full distribution
   - There are 2^d entries in the d-dimensional distribution
   - Then aggregate results here (obtaining additional error)

- Still using randomized response on each entry
  - Approach 1 (marginals): cost proportional to $2^{3k/2} d^{k/2}/\sqrt{N}$
  - Approach 2 (full): cost proportional to $2^{(d+k)/2}/\sqrt{N}$

THE UNIVERSITY OF WARWICK

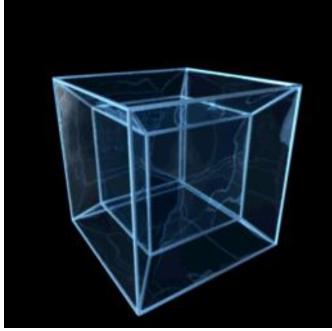
What to materialize?

Different approaches based on how information is revealed

1. We could reveal information about all marginals of size k
   - There are (d choose k) such marginals, of size 2^k each
2. Or we could reveal information about the full distribution
   - There are 2^d entries in the d-dimensional distribution
   - Then aggregate results here (obtaining additional error)

- Still using randomized response on each entry
  - Approach 1 (marginals): cost proportional to $2^{3k/2} d^{k/2}/\sqrt{N}$
  - Approach 2 (full): cost proportional to $2^{(d+k)/2}/\sqrt{N}$
- If k is small (say, 2), and d is large (say 10s), Approach 1 is better
  - But there's another approach to try…

Hadamard transform

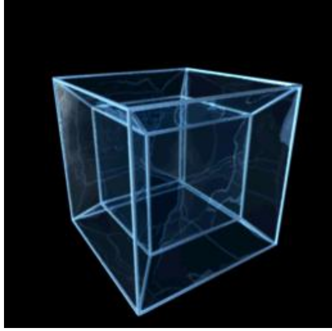


Instead of materializing the data, we can transform it

- Via Hadamard transform (the discrete Fourier transform for the binary hypercube)
  - Simple and fast to apply

$$\begin{bmatrix} H^* & H^* \\ H^* & -H^* \end{bmatrix} = \begin{bmatrix} -1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 \\ -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 \\ 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 \\ 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 \\ 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 \end{bmatrix}.$$

Hadamard transform



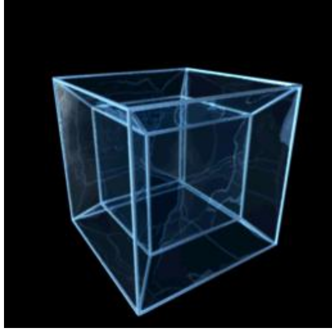
Instead of materializing the data, we can transform it

- Via Hadamard transform (the discrete Fourier transform for the binary hypercube)
  - Simple and fast to apply

$$\begin{bmatrix} H^* & H^* \\ H^* & -H^* \end{bmatrix} = \begin{bmatrix} -1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 \\ -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 \\ 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 \\ 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 \\ 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 \end{bmatrix}.$$

- Property 1: only (d choose k) coefficients are needed to build any k-way marginal
  - Reduces the amount of information to release

Hadamard transform



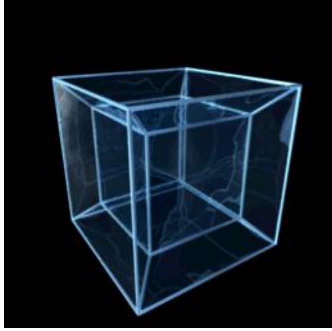
Instead of materializing the data, we can transform it

- Via Hadamard transform (the discrete Fourier transform for the binary hypercube)
  - Simple and fast to apply

$$\begin{bmatrix} H^* & H^* \\ H^* & -H^* \end{bmatrix} = \begin{bmatrix} -1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 \\ -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 \\ 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 \\ 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 \\ 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 \end{bmatrix}.$$

- Property 1: only (d choose k) coefficients are needed to build any k-way marginal
  - Reduces the amount of information to release
- Property 2: Hadamard transform is a linear transform
  - Can estimate global coefficients by sampling and averaging

Hadamard transform



Instead of materializing the data, we can transform it

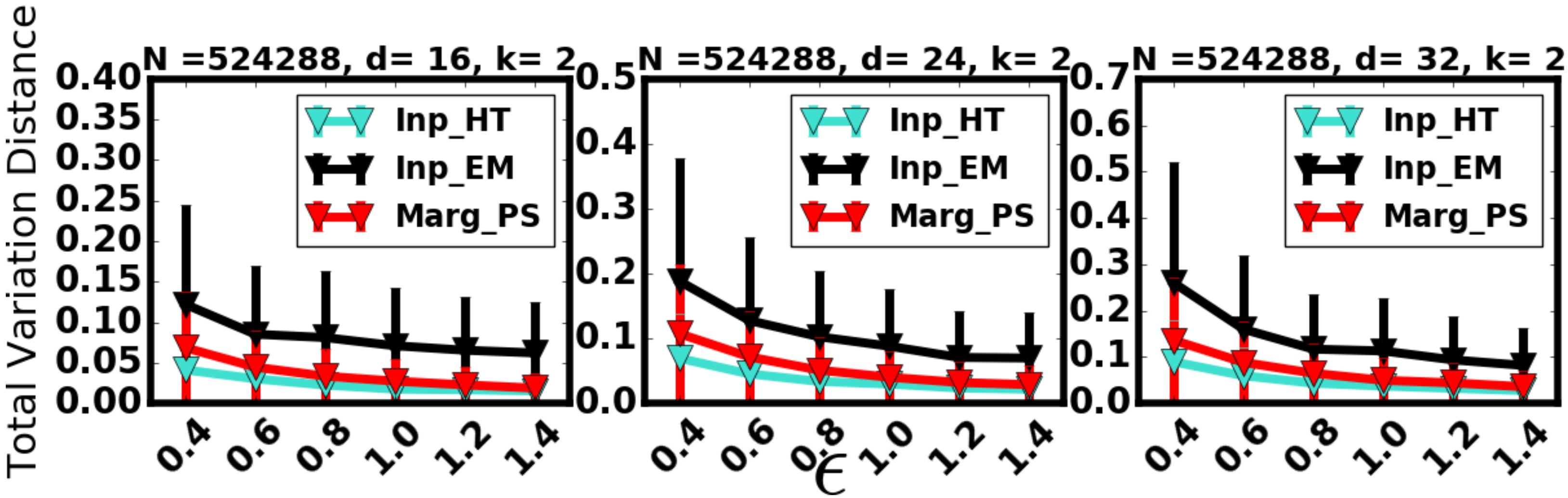
- Via Hadamard transform (the discrete Fourier transform for the binary hypercube)
  - Simple and fast to apply

$$\begin{bmatrix} H^* & H^* \\ H^* & -H^* \end{bmatrix} = \begin{bmatrix} -1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 \\ -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 \\ 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 \\ 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 \\ 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 \end{bmatrix} .$$

- Property 1: only (d choose k) coefficients are needed to build any k-way marginal
  - Reduces the amount of information to release
- Property 2: Hadamard transform is a linear transform
  - Can estimate global coefficients by sampling and averaging
- Yields error proportional to $2^{k/2}d^{k/2}/\sqrt{N}$
  - Better than both previous methods (in theory)

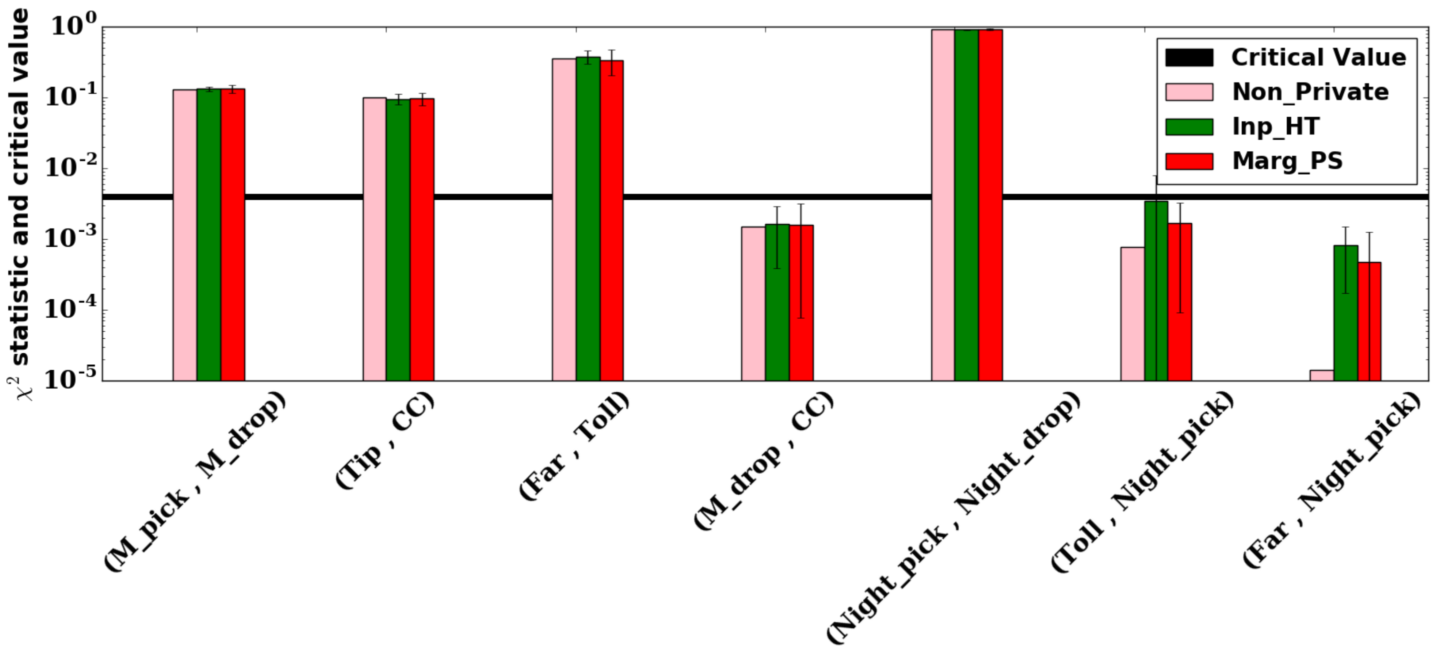
THE UNIVERSITY OF WARWICK

Empirical behaviour



- Compare three methods: Hadamard based (Inp_HT), marginal materialization (Marg_PS), Expectation maximization (Inp_EM)
- Measure sum of absolute error in materializing 2-way marginals
- N = 0.5M individuals, vary privacy parameter ε from 0.4 to 1.4

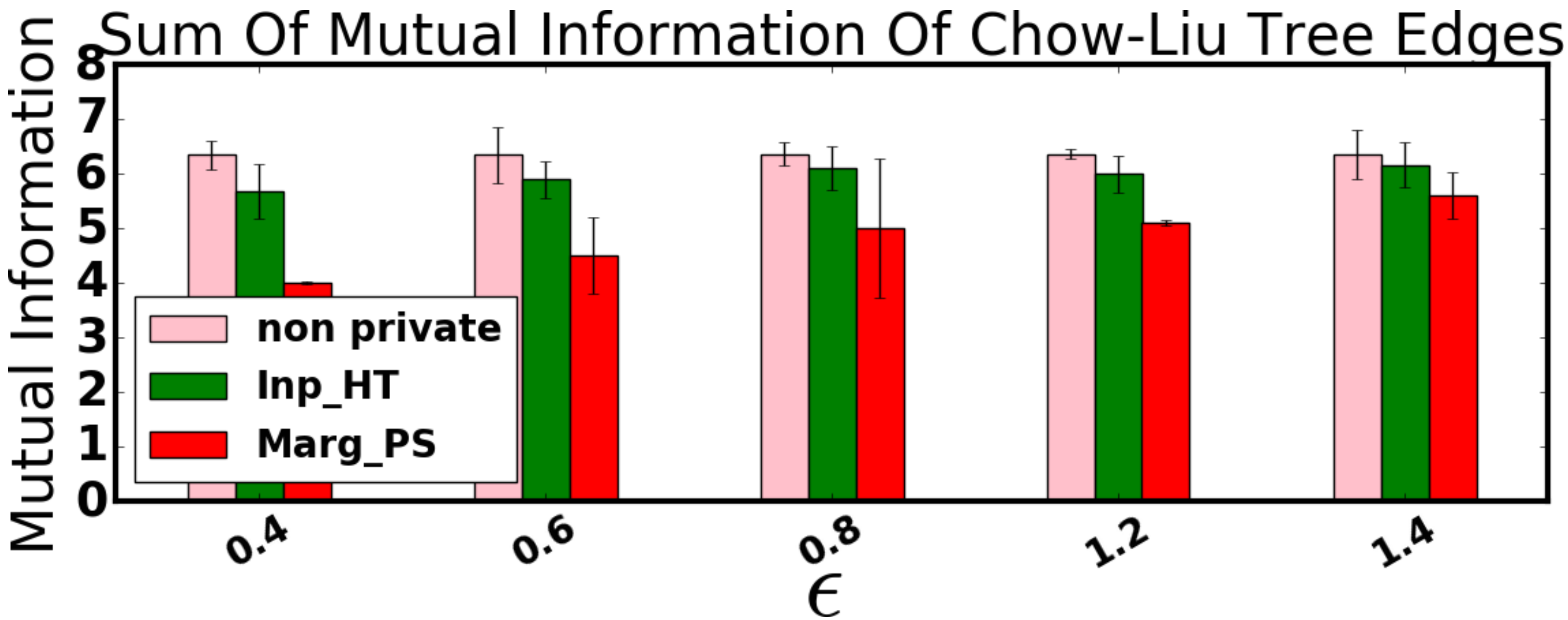
Applications – χ-squared test



- Anonymized, binarized NYC taxi data
- Compute χ-squared statistic to test correlation
- Want to be same side of the line as the non-private value!

THE UNIVERSITY OF WARWICK

Application – building a Bayesian model



- Aim: build the tree with highest mutual information (MI)
- Plot shows MI on the ground truth data for evaluation purposes

THE UNIVERSITY OF WARWICK