Private and Efficient Federated Numerical Aggregation

Graham Cormode, Igor L. Markov, Harish Srinivas

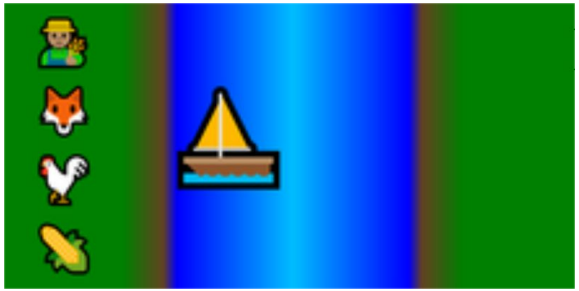
Overview

- The challenges for large scale private data analysis
- Private mean estimation as a fundamental task
- Our approach: bit-pushing
- Experimental evaluations and deployment experience

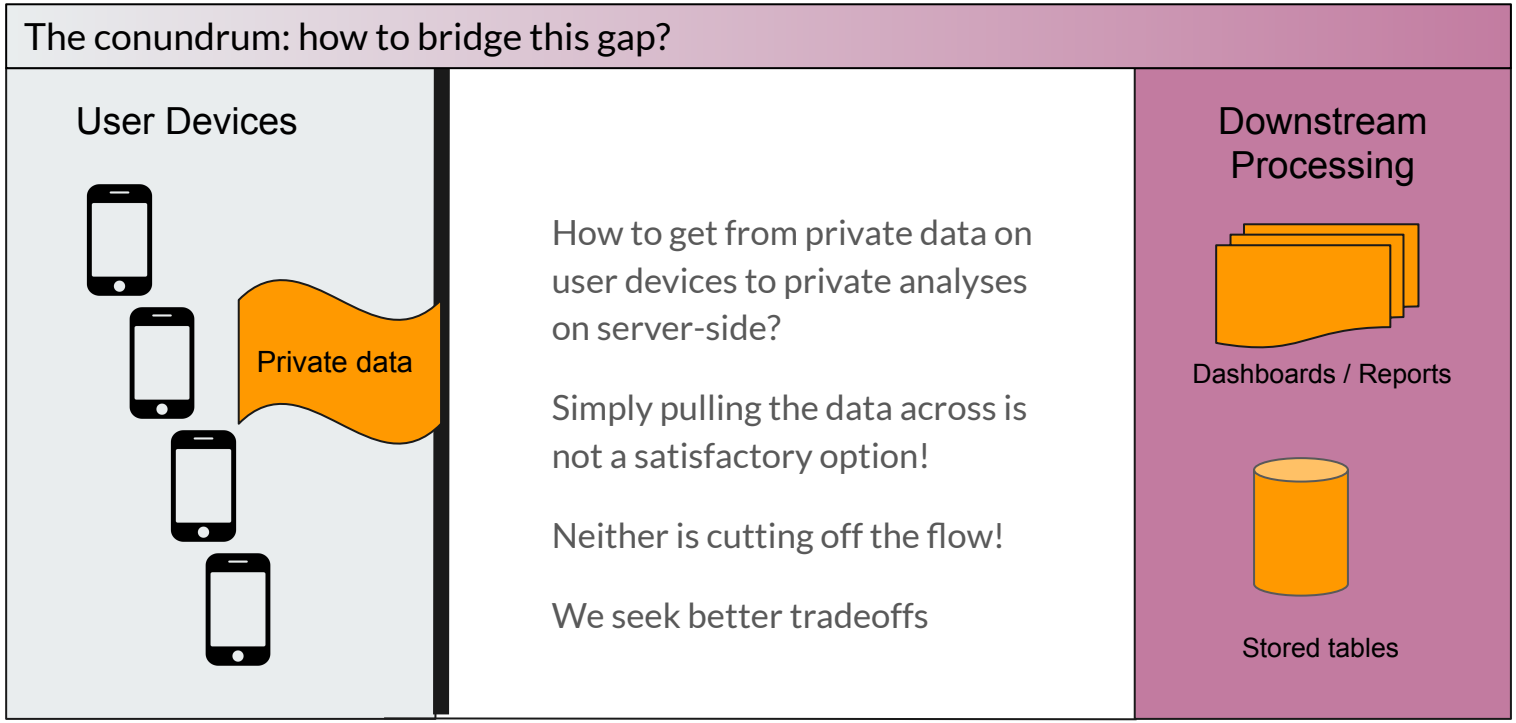
Images are sourced from Wikimedia Commons and attributed accordingly in the notes.

The Private Data Analysis conundrum

- Many large companies have been built on the basis of analyzing data from many users
  - E.g., online advertising, need to understand consumer interests, and provide tailored advertising
- Technology ecosystem changes, new regulations and sensitivity to privacy concerns affect data flow
  - E.g., Apple opt-in data sharing, GDPR/ePD, privacy preservation as a feature
- Conundrum: to understand (sub)population behaviour without compromising individual privacy?
- Canonical example: analyzing user actions in apps on personal devices to central servers



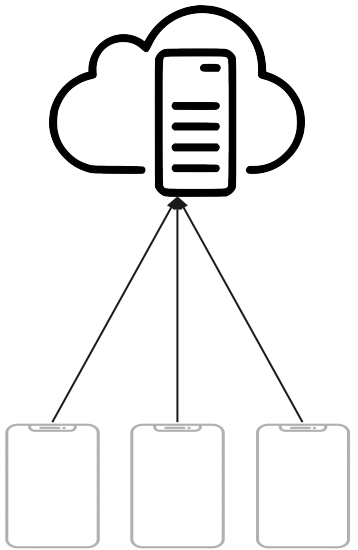
The conundrum: how to bridge this gap?



The Federated Approach

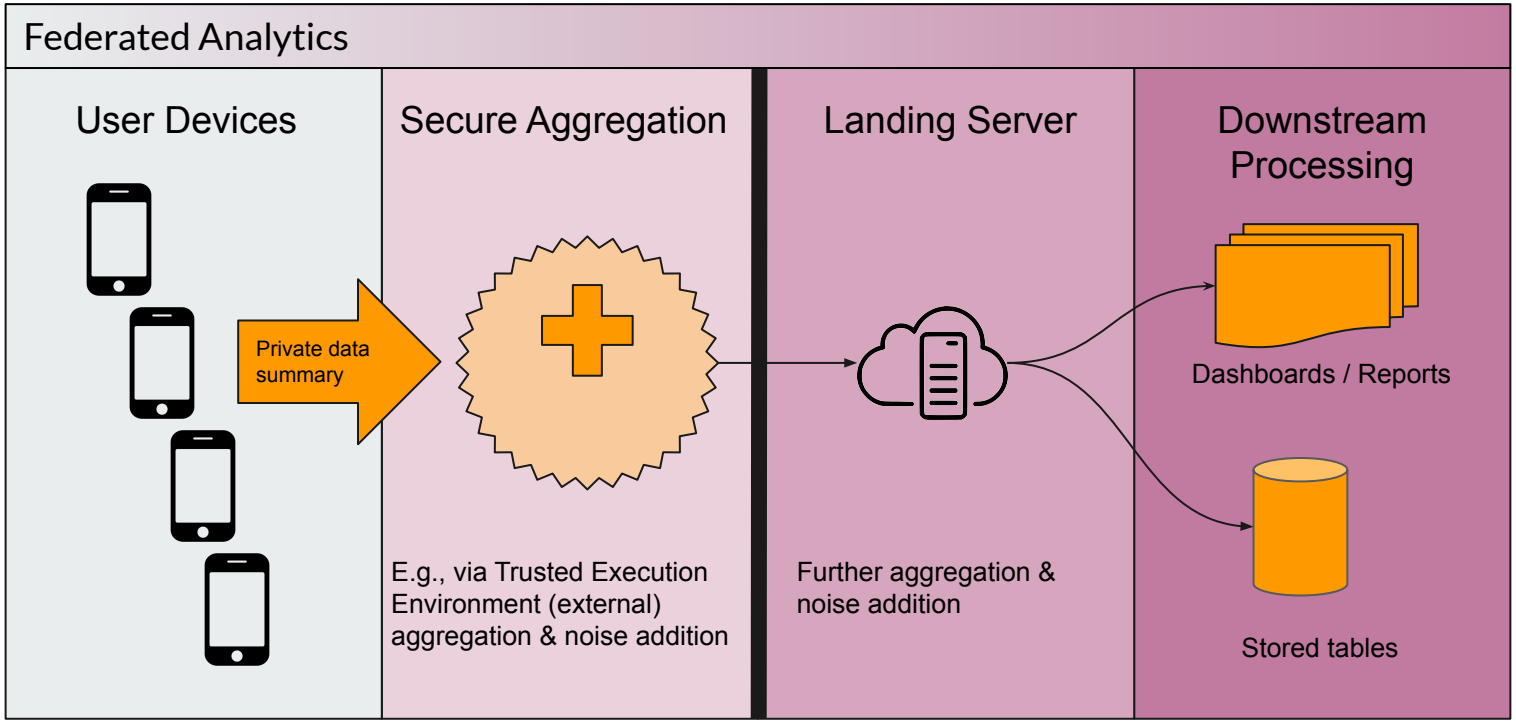
The federated approach to computation aims to support privacy requirements:

- Data remains under the control of user *clients* (e.g., on their phone)
- Only a small amount of necessary data is shared with central *servers*
- Communication is done under strong security guarantees (e.g., encryption)
- Additional privacy guarantees are provided (e.g., via adding random noise, anonymous communication channels, secure aggregation etc.)



So federated computation is **secure**, **private** and **distributed** (but not fully decentralized)

This builds on prior work that achieves subsets of these properties, and is becoming popular in industry


Federated Analytics
User Devices
Secure Aggregation
Landing Server
Downstream Processing
Private data summary
E.g., via Trusted Execution Environment (external) aggregation & noise addition
Further aggregation & noise addition
Dashboards / Reports
Stored tables

Mean and variance estimation with privacy

A common business need: measure the mean and variance of some quantity of interest

Requirements:

- A simple algorithm that can be executed on low-powered hardware (phones, wearables)
- Privacy guarantees that are meaningful to both experts and non-experts alike
- Accurate results that can be applied over large (thousands, millions, billions) populations

1-bit Privacy framework

We want to offer simple user-friendly privacy promises:

- Your data stays on device, the server only learns aggregated statistics
- Your device only reports a single bit of information (a zero or a one)
- The reported bit is noisy: it is perturbed with some probability (randomized response)

This is backed up by more formal privacy and security guarantees:

- The algorithm performed using Federated Analytics via Secure Aggregation
- The data sent is perturbed with (Local) differential privacy

Basic bit-pushing algorithm

Each client holds input x as a b-bit integer

Server samples j in [1..b], asks client to share x[j]

Server computes the mean of each bit location

Server reports the weighted sum of bit means

Theory: can analyze variance of this procedure (leads to sampling bit j with probability $2^{\alpha j}$)

Additional privacy: apply randomized response

Algorithm 1: Basic bit-pushing algorithm

Input : No. of bits b, bit weights p, no. of clients n
Output: (Result r, mean of bits m, sum of bits s)

Initialize result $r = 0$
for $j = 0$ **to** $b - 1$ **in parallel do**
  Contact $c[j] = p[j] \cdot n$ clients to request bit j
  Gather weighted sum of bits as $s[j]$
  Compute bit means $m[j] = s[j]/c[j]$
  $r \leftarrow r + (2^j \cdot m[j])$
return (r, m, s)

Adaptive Bit-pushing

A more advanced solution uses two rounds of bit-pushing:

- Round 1: figure out which bits matter most
- Round 2: gather more data on the important bits
- Combine all reports to make final estimate

Theory: can bound variance of final estimator

Privacy: also apply randomized response here

Extra accuracy: downweight bits that look like noise ("bit squashing")

Algorithm 2: Adaptive bit-pushing algorithm

Input : No. of bits b, no. of clients n, parameters α, γ, δ
Output : Result r

/* Round 1: */
for $j = 0$ **to** $b - 1$ **do**
  Compute $p_1[j] = (2^j)^\gamma$
Normalize p_1: $p_1 \leftarrow p_1 / \text{sum}(p_1)$
Run basic bit-pushing:
  $(r_1, m_1, s_1) = \texttt{BitPushing}(b, p_1, \delta n)$
/* Round 2: */
for $j = 0$ **to** $b - 1$ **do**
  Compute $p_2[j] = (4^j \cdot m_1[j] \cdot (1 - m_1[j])^\alpha$
Normalize p_2: $p_2 \leftarrow p_2 / \text{sum}(p_2)$
Run basic bit-pushing:
  $(r_2, m_2, s_2) = \texttt{BitPushing}(b, p_2, (1 - \delta)n)$
/* Final aggregation: */
Combine means $m_3 = (s_1 + s_2)/(\delta n * p_1 + (1 - \delta)n * p_2)$
for $j = 0$ **to** $b - 1$ **do**
  $r \leftarrow r + 2^j \cdot m_3[j]$
return r

Extensions

Weighted sampling: bit j is sampled with probability p_j proportional to $2^{\alpha j}$

- Our theoretical analysis suggests $\alpha = 0.5$ is optimal, we test with $\alpha = 0.5$ and $\alpha = 1$

Estimating variance: we can estimate variance by transforming the input values (carefully)

Other functions: higher moments, products, geometric means etc. can also be estimate via bit-pushing

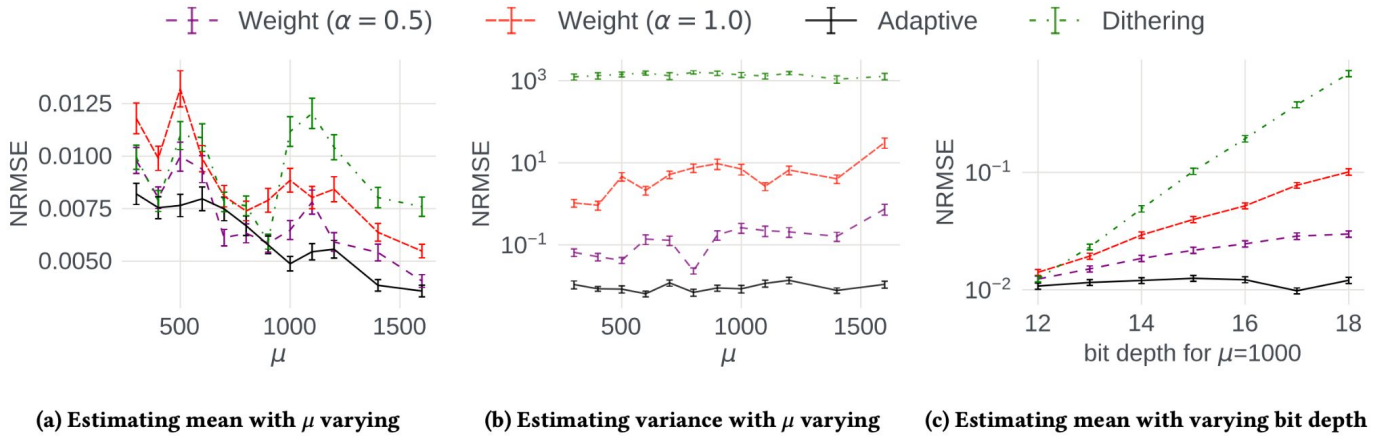
Privacy models: can obtain central DP guarantees via distributed noise addition or secure hardware

Trust issues: who should sample the bit? It's more robust if the server nominates which bit to report

Pre-deployment experimental study

- Initial experiments on mixture of real and synthetic (normal) data to choose parameter settings
- We will vary:
  - Input distribution mean μ
  - Function being estimated: mean or variance
  - Number of bits n_bits the algorithms operate over
  - Size of population N participating in the data collection
- Results averaged over 100 repetitions
- Compared against a competitor method Subtractive Dithering: randomized rounding + randomized response
  - See [Ben Basat, Mitzenmacher, Vargaftik ICALP 2021]
  - We also compared several more (see paper) which were worse than Subtractive Dithering, so we omit these
- After fixing the parameters, we report on deployment experience in a production environment

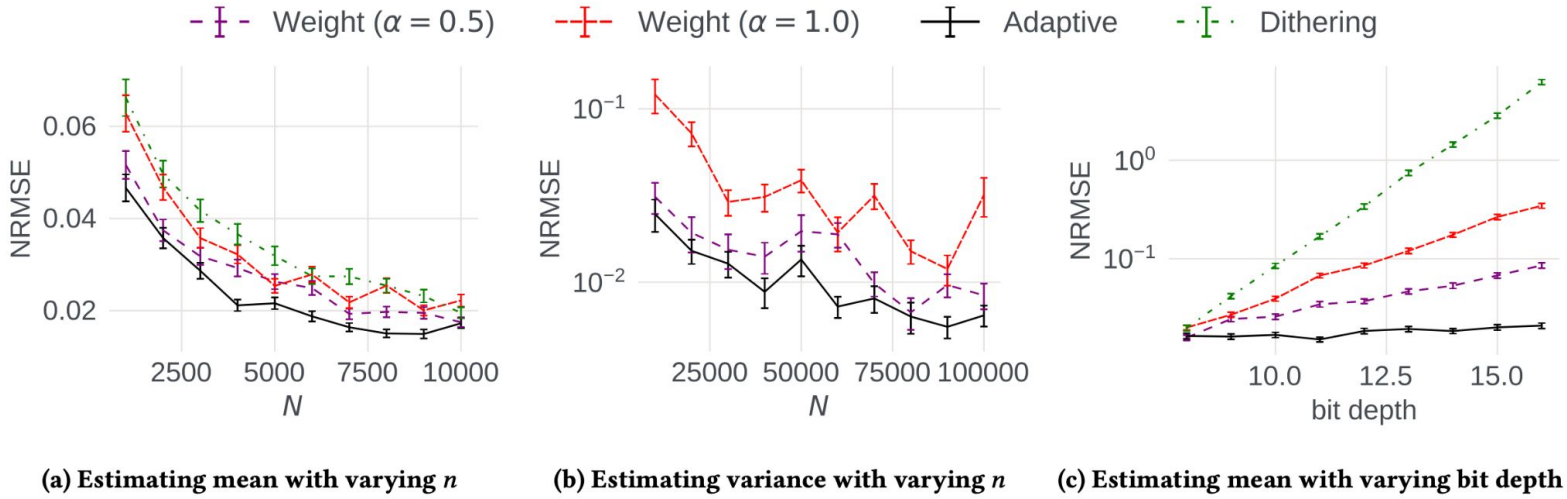
Experiments on synthetic Normal data



(a) Estimating mean with μ varying **(b) Estimating variance with μ varying** **(c) Estimating mean with varying bit depth**

Adaptive (2 round) approach is generally preferred. Advantage improves as number of bits increases

Experiments on real Census dataset



(a) Estimating mean with varying n **(b) Estimating variance with varying n** **(c) Estimating mean with varying bit depth**

Similar behavior on real data. Variance estimation is harder to get good accuracy, need more clients

Deployment experience

Bit-pushing has been evaluated in production on device health statistics. Many observations:

- Small number of bits (8-16) is good enough to obtain sufficiently accurate results
- Local differential privacy introduces noise, but the message is still clear
- Central differential privacy introduced by trusted hardware is preferred
- Client dropouts are common, but do not impact on accuracy: dropouts are not correlated with data
- Results can be gathered quickly, in a matter of minutes from a large enough population
- There is a lot of heterogeneity across multiple aspects: some clients have many values, some only 1
    - Sampling or averaging locally can be used, and this does not appreciably affect accuracy

More detailed observations and discussion in the paper

Concluding Remarks

- Bit-pushing seems viable, and preferable to naive noise addition
  - Adaptive bit-pushing is effective, and most accurate when not applying DP
  - Bit squashing can control the error as more bits are added
- The main advantage comes due to ability to “zoom in” on the range of data
  - For variables with stable behaviour, the behavior is more similar to other options
- The approach is fairly robust to poisoning
  - It is hard to arrange client responses to distort the value that is computed
- Bottom line: real-world systems need more simple algorithms with strong privacy guarantees