Consistency with External Knowledge: The TopDown Algorithm

Daniel Kifer

Simons Privacy Workshop

TopDown algorithm developers: Robert Ashmead, Simson Garfinkel, Philip Leclerc, Brett Moran, William Sexton, Pavel Zhuravlev

Academic collaborators: Michael Hay, Ashwin Machanavajjhala, Gerome Miklau
All opinions, statements, conclusions, etc., in this talk are my own (as a researcher on differential privacy), and are not the official position of the U.S. Census Bureau.
1. Introduction

2. Schema Extension: TopDown without invariants

3. Invariants

4. The TopDown Algorithm with invariants

5. zCDP/RDP vs. Pure DP
Introduction

Goal

- DAS: disclosure avoidance system
- Publish a histogram with billions of cells using formal privacy.
  - Location (hierarchical) - National, State, County, Tract, Block Group, Block. \( \approx 6 \text{ million blocks} \)
  - Ethnicity: 2 values
  - Race: 63 values
  - Voting age: 2 values
  - Residence type ("household" or group quarters code) - 8 values

Hierarchical workload

- Counting queries about demographics in each geographic region
- E.g., 2010 PL94-171 Redistricting and Advanced Group Quarters Summary Files

The data are sparse

- \( \approx 12 \text{ billion cells} \)
- \( \approx 309 \text{ million people} \)
- Workload: 641 non-identity queries per geo-unit \( \approx 3.6 \text{ billion queries} \)
- +12 billion identity queries
Differential Privacy

**Definition (Differential Privacy (DMNS06))**

Let $\epsilon > 0$. An algorithm $M$ satisfies $\epsilon$-differential privacy if for all $\omega \in \text{range}(M)$ and all pairs of databases $D_1, D_2$ that differ on the value of one page of Census questionnaire (information about 1 person),

$$P(M(D_1) = \omega) \leq e^\epsilon P(M(D_2) = \omega)$$

- Note: multiple tables
- Person demographics: 1 person affects 1 row.
- Households/Housing units: 1 person can modify 1 row in a bounded way (different from Uber’s model)
- Group Quarters: similar to households
- Geographic boundaries: no protection
Introduction

Requirements

- Create microdata
  - Ensures that published “universe person” tabulations are mutually consistent.
  - Also system requirement: output of DAS goes into tabulation system.
  - Equivalent to histogram with nonnegative integer entries.

- Run within X days
  - Implemented in Spark
  - Uses GovCloud
  - Use commercial-grade optimizers (e.g., Gurobi, CPLEX)

- Run before all data are available
  1. PL94-171 first
  2. Summary File 1
  3. Urban/Rural update
  4. etc.

- Consistent with external pieces of knowledge
- Consistent with prior releases
Some datasets are treated as effectively public.
- Local Update of Census Addresses Operation (LUCA) dataset contains number of housing units and GQ units of each type in each block.
- Number of occupied GQ facilities of each type in each block assumed to be known.

Some information might be declared public as policy decision.
- In 2010: population of each block.
- In 2010: number of occupied housing units in each block
  - \# occupied housing units = \# of householders

Invariants:
- Queries in true data that must have same answers in “privatized” data.
- Differentially private algorithms are still differentially private.
- Privacy semantics, however, are awkward.
- Easily make simple problems NP hard.

Structural zeros:
- Data-independent restrictions
  - 0 householders aged 14 and under
  - \# householders \geq \# spouses + \# unmarried partners of householders.
Invariants and Utility

- Invariants may be forced by policy decisions.
- Invariants based on external knowledge can increase trust in the microdata.
- Utility:
  - Making published data consistent with the invariants could increase accuracy of microdata.
  - In experiments, feasible datasets (satisfying invariants) can be very different from unrestricted datasets (given the same noisy measurements).
Introduction

The Spherical Cows

- Incremental Schema Extension - Incrementally add columns to DP microdata
  - e.g., start with Race (R), Ethnicity (E), Voting Age status (VA)
  \[
  \begin{array}{ccc}
  R & E & VA \\
  \rightarrow & R & E & VA & State \\
  \rightarrow & R & E & VA & State & County \\
  \end{array}
  \]
  \[\rightarrow \text{ etc.}\]
  - Necessary because not all data are available at once.
  - Also useful for scalability.
  - Microdata generation: measure then postprocess
  - Cannot fit postprocessing optimization problem in memory

- Consistency with External Knowledge
  - Linear constraints on histogram constructed from full schema.
  - Ensure there exists an extension of \( R \ E \ VA \) that will satisfy those constraints.
  - Decision problem (microdata are consistent?) is NP complete.
Outline

1. Introduction
2. Schema Extension: TopDown without invariants
3. Invariants
4. The TopDown Algorithm with invariants
5. zCDP/RDP vs. Pure DP
Histogram is too big to fit in memory, must be created in pieces.

First generate nonnegative integer histogram $H$ at the national level.

Create child histograms $H_i$ for each state $S_i$, with $\sum_i H_i = H$.

Recursively create county, tract, block group, block level histograms.

Number of optimization problems increases down the hierarchy

Size of optimization problems decreases

- Algorithm estimates which counts are nonzero
- Splits these counts among children
- Variables that are 0 at the parent are dropped from future optimizations.
Total U.S. population is not protected.

Given linear query workload $W$, use High-dimensional matrix mechanism to obtain [MMHM2018] linear queries $Q$ to ask.

Obtain noisy measurements $M = Q(H) + \text{Noise}$

Solve $H^* = \arg \min_{H^*} \| Q(H^*) - M \|_2^2 \text{ s.t. } \text{sum}(H^*) = n \text{ and } H^* \succeq 0$

Now we have a nonegative fractional histogram of population demographics.
National Histogram Linear solve

- Nonnegative fractional histogram $H^*$.
- Round using LP

$$\arg \min_{\tilde{H}} \| \tilde{H} - H^* \|_1$$

s.t. $\tilde{H} \succeq 0$ (nonnegativity)

$$|\tilde{H}[x] - H^*[x]| \leq 1 \text{ for all cells } x$$

$$\sum_x \tilde{H}[x] = \sum_x H^*[x] \text{ (total sum constraint)}$$

- Constraint matrix is **Totally Unimodular** (TUM).
- Many LP algorithms (barrier+crossover, simplex) give integer solutions.
- To be safe, implementation asks Gurobi to solve IP instead of LP (fast because of TUM)
Now we have a nonnegative integer histogram $\tilde{H}$
- National level demographics
- Equivalent to microdata with no geography

Next we add States + DC.
- $H_i$: demographics histogram for state $i$
  - Ignore cells that are 0 at national level DP histogram $\tilde{H}$
  - Reduces size of the optimization problem.
- Given workload at each state + DC, use HDMM to obtain linear queries $Q$ to ask.
- Noisy measurement for state $i$: $M_i = Q(H_i)+\text{Noise}$
- Then we solve an $L_2$ followed by $L_1$ optimization problem.
\( \tilde{H} \) is national level DP histogram

Noisy state level measurements \( M_1, \ldots, M_{51} \)

Obtain DP state-level nonnegative fractional histograms that add up to \( \tilde{H} \)

\[
\arg \min_{H_1^*, \ldots, H_m^*} \sum_{j=1}^{m} \| Q(H_j^*) - M_j \|_2^2
\]

s.t. \( H_j^* \geq 0 \) for all \( j \)

\[
\sum_{j=1}^{m} H_j^* = \tilde{H}
\]
Now round using IP that is equivalent to LP when using e.g., barrier+crossover or simplex algorithms.

\( H_j^* \) are nonnegative fractional state level histograms

\[
\arg \min_{\tilde{H}_1, \ldots, \tilde{H}_m} \sum_{j=1}^{m} \| \tilde{H}_j - H_j^* \|_1
\]

s.t. \( \tilde{H}_j \succeq 0 \) for all \( j \)

\[ |\tilde{H}_j[x] - H_j^*[x]| \leq 1 \) for all \( j \) and cells \( x \)

\[ \sum_j \tilde{H}_j = \tilde{H} \]
(In parallel) For each state, we generate its county level histograms.
For each county, generate its tract histograms.
For each tract, generate its block level histograms.
Convert back to microdata.
≈ 20k lines of code
≈ 60k more lines of supporting code
TopDown Algorithm

BUT WAIT, THERE'S MORE!
Final data (with all fields) must satisfy (mostly) linear constraints.

Consumed most time & effort.

- Semantics:
  - What is impact on privacy if some exact statistics about data are published?
  - How do privacy semantics change?
  - Needed for policy decisions.
  - Short answer: it’s complicated.

- Algorithm:
  - How do we enforce them in DP microdata?
  - Short answer: it’s complicated.
Small college town, 2 regions

Every student lives in dorms
  - Male-only (M)
  - Female-only (F)
  - Co-ed (C)

Knowledge:
  - 100 students in each region:
    \[ F_1 + C_1 + M_1 = F_2 + C_2 + M_2 = 100 \]
  - All dorms are occupied.
  - \( R_1 : 0 \) Male, 1 Female, 1 Co-ed dorms:
    \[ M_1 = 0; F_1 \geq 1; C_1 \geq 1. \]
  - \( R_2 : 1 \) Male, 0 Female, 1 Co-ed dorms:
    \[ M_2 \geq 1; F_2 = 0; C_2 \geq 1. \]

We already generated town-wide DP statistics: \( \tilde{F}, \tilde{C}, \tilde{M} \).

Consistent with background knowledge?
Knowledge:

- 100 students in each region:
  \[ F_1 + C_1 + M_1 = F_2 + C_2 + M_2 = 100 \]
- All dorms are occupied.
- \( R_1 \): 0 Male, 1 Female, 1 Co-ed dorms:
  \[ M_1 = 0; F_1 \geq 1; C_1 \geq 1. \]
- \( R_2 \): 1 Male, 0 Female, 1 Co-ed dorms:
  \[ M_2 \geq 1; F_2 = 0; C_2 \geq 1. \]

Consistency: implications for \( \tilde{F}, \tilde{C}, \tilde{M} \)?
Knowledge:

- 100 students in each region:
  \[ F_1 + C_1 + M_1 = F_2 + C_2 + M_2 = 100 \]
- All dorms are occupied.
- \( R_1 : 0 \) Male, 1 Female, 1 Co-ed dorms:
  \[ M_1 = 0; F_1 \geq 1; C_1 \geq 1. \]
- \( R_2 : 1 \) Male, 0 Female, 1 Co-ed dorms:
  \[ M_2 \geq 1; F_2 = 0; C_2 \geq 1 \]

Consistency: implications for \( \tilde{F}, \tilde{C}, \tilde{M} \)?

- \( \tilde{M} \geq 1 \)
- \( \tilde{F} \geq 1 \)
- \( \tilde{C} \geq 2 \)
- \( \tilde{F} + \tilde{C} + \tilde{M} = 200 \)
- Are we done?
Knowledge:
- 100 students in each region:
  \[F_1 + C_1 + M_1 = F_2 + C_2 + M_2 = 100\]
- All dorms are occupied.
- \(R_1\): 0 Male, 1 Female, 1 Co-ed dorms:
  \[M_1 = 0; F_1 \geq 1; C_1 \geq 1.\]
- \(R_2\): 1 Male, 0 Female, 1 Co-ed dorms:
  \[M_2 \geq 1; F_2 = 0; C_2 \geq 1.\]

Consistency: implications for \(\sim F, \sim C, \sim M\)?
- \(\sim M \geq 1, \sim F \geq 1, \sim C \geq 2,\)
  \[\sim F + \sim C + \sim M = 200, \quad ??\]

Suppose \(\sim F = 49, \sim C = 50, \sim M = 101\)
- Satisfies these constraints
- But, only 1 male-only dorm.
- It is in region with 100 students.
- \(\therefore \sim M = 101\) is not valid
An Example (5)

Knowledge:

- 100 students in each region:
  \[ F_1 + C_1 + M_1 = F_2 + C_2 + M_2 = 100 \]
- All dorms are occupied.
- \( R_1 \): 0 Male, 1 Female, 1 Co-ed dorms:
  \( M_1 = 0; \ F_1 \geq 1; \ C_1 \geq 1 \).
- \( R_2 \): 1 Male, 0 Female, 1 Co-ed dorms:
  \( M_2 \geq 1; \ F_2 = 0; \ C_2 \geq 1 \).

Consistency: implications for \( \tilde{F}, \tilde{C}, \tilde{M} \)?

The necessary and sufficient constraints (auto-proved via FME):

\[
\begin{align*}
\tilde{F} & \geq 1 \\
\tilde{C} & \geq 2 \\
\tilde{M} & \geq 1 \\
\tilde{F} & \leq 99 \\
\tilde{C} + \tilde{F} & \geq 101 \\
\tilde{C} + \tilde{F} + \tilde{M} & = 200
\end{align*}
\]
Invariants via Network Flows

- Reduction to Network Flow (change $\geq c$ constraints to $\geq 0$)
- Use max-flow/min-cut theorem
Starting schema: $S_0$ (set of table columns)
  - e.g., \{Dorm Type\}
Extended schema $S \supset S_0$
  - e.g., \{Dorm Type, Region\}
$T_0$: microdata table with schema $S_0$
$T$: microdata table with schema $S$
$C$: set of constraints on $T$
  - Total population in each region
  - Presence/absence of occupied dorms
$C_0$: set of constraints on $T_0$
  - What we want
  - Constraints on population in each dorm in $T_0$
Invariants

Implied constraints

**Definition (Necessary Constraints)**

$C_0$ is necessary if $C(T) = \text{true} \Rightarrow C_0(T_0) = \text{true}$, where $T_0$ is projection of $T$ onto the attributes in schema $S_0$

**Definition (Sufficient Constraints)**

$C_0$ is sufficient if $C_0(T_0) = \text{true} \Rightarrow$ there exists an extension $T$ of $T_0$ with $C(T) = \text{true}$

We want $C_0$ to be necessary and sufficient:

- $\tilde{T}_0$: DP microdata
- Sufficient: If $C_0(\tilde{T}_0) = \text{true}$, we can always add columns to get a DP version $\tilde{T}$ that satisfies $C$
- Necessary: Constraints are not too restrictive (do not add unnecessary bias)
How do we find them?
NP-complete in universe size when $|S_0| = 2$ and $|S| = 3$. Easily encodes 3-SAT
NP-complete if each region only has equality constraints for 2 one-way marginals
- NP-complete in # of regions and size of one of the marginals (if 2nd marginal has size 3)

### Region A

<table>
<thead>
<tr>
<th>$R_H$</th>
<th>$R_V = 0$</th>
<th>$R_V = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_H = 0$</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>$R_H = 1$</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>5</td>
</tr>
</tbody>
</table>

### Region B

<table>
<thead>
<tr>
<th>$R_H$</th>
<th>$R_V = 0$</th>
<th>$R_V = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_H = 0$</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>$R_H = 1$</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>15</td>
</tr>
</tbody>
</table>

But exists an inefficient algorithm if constraints are linear:
- Fourier-Motzkin elimination (FME).
- Double-exponential complexity (Can be accelerated but not for our scale)
- Works for fractional histograms (often provable for integer histograms).
Outline

1. Introduction
2. Schema Extension: TopDown without invariants
3. Invariants
4. The TopDown Algorithm with invariants
5. zCDP/RDP vs. Pure DP
State Level Histograms: $L_2$ solve with invariants

- $\tilde{H}$ is national level DP histogram
- Compute implied constraints $C_i$ for each state $i$
- Noisy state level measurements $M_1, \ldots, M_{51}$
- Obtain DP state-level nonnegative fractional histograms that add up to $\tilde{H}$

$$\arg \min_{H_1^*, \ldots, H_m^*} \sum_{j=1}^{m} \|Q(H_j^*) - M_j\|_2^2$$

s.t. $H_j^* \succeq 0$ for all $j$

$$C_i(H_j^*) = \text{true} \quad \text{for all } j$$

$$\sum_{j=1}^{m} H_j^* = \tilde{H}$$
This rounding using IP that is equivalent to LP when using barrier+crossover or simplex algorithms.

- Under conditions like TUM constraint matrix or nice obj + rhs
- $H^*_j$ are nonnegative fractional state level histograms

$$\arg\min_{\tilde{H}_1, \ldots, \tilde{H}_m} \sum_{j=1}^{m} ||\tilde{H}_j - H^*_j||_1$$

s.t. $\tilde{H}_j \succeq 0$ for all $j$

$|\tilde{H}_j[x] - H^*_j[x]| \leq 1$ for all $j$ and cells $x$

$C_i(\tilde{H}_j) = true$ for all $j$

$$\sum_j \tilde{H}_j = \tilde{H}$$
Implied constraints deduced by hand + FME

\( L_2 \) solve: creates nonnegative fractional histogram
- Implied constraints \( C_0 \) are added to the problem.
- Implies fractional feasible extension exists.

\( L_1 \) solve: rounds to nonnegative integer counts.
- Generally, linear implied constraints do not always guarantee feasible integer solution
- They do if the problem constraint matrix is TUM (then linear solve is also usually fast)
- Some of our implied invariant constraints are not TUM
  - But integer optimal solution exists
  - Solve is slow
  - Possibly equivalent to TUM constraints (network flow and a few others)
3 digit GQ code of occupied group quarters might be invariant
- Similar to college dorm example
- But 28 types of GQ
- In general, \( \approx 2^{28} \) implied constraints, one for each combination of GQ.
- Can be much smaller, depending on data.
- For each combination \( S \) of GQ:
  - Total population living in GQ of types in \( S \) is \( \leq c \)
  - \( c \) depends on total population in blocks that have GQ types from \( S \)
- Constraint matrix is not TUM
  - Might be equivalent to TUM (via network flows)
  - Network flow integrality theorem says an integer solution exists
Workarounds

- "The Failsafe"
  - In the worst case, breaks out of the framework.
  - If a solve fails (or is slow) in, e.g., county level histogram $H_c$
    - Cannot find feasible tract histograms $H_1, \ldots, H_k$ with $\sum_i H_i[x] = H_c[x]$ for all $x$
    - Drop this requirement
    - Use weaker requirements (e.g., total population matches: $\sum_i \sum_x H_i[x] = \sum_x H_c[x]$) and other tricks
    - Generate tracts
    - The county is changed to the sum of the tracts
    - Worse accuracy but invariants maintained

- "Minimal Schema"
  - $S_0$: smallest set of attributes that cover the invariants + all geography.
  - Generate nonnegative integer histogram in 2 solves $L_2$ followed by $L_1$.
    - Simultaneously for all levels of geography, estimate group quarters population by GQ type (nothing else)
  - Then extend to the other attributes.
  - Works if these problems fit in memory
  - Cutting plane: find the instance-level necessary constraints

(Simons Privacy Workshop)
The TopDown Algorithm with invariants

Current Invariants

- Have explored many invariants.
- Choice of invariants is policy decision.
  - Policy can be affected by privacy semantics
  - Policy can be affected by computational difficulty
- Current set of invariants being explored:
  - State population totals are invariant.
  - # occupied GQ facilities of each type in each block are invariant.
  - Total # of housing units in each block are invariant.
  - Auxiliary information about GQ (age restrictions, female-only, male-only, co-ed).
  - Also structural zeros.
- Historical invariants deducible from
Outline

1. Introduction
2. Schema Extension: TopDown without invariants
3. Invariants
4. The TopDown Algorithm with invariants
5. zCDP/RDP vs. Pure DP
Currently using pure DP with Laplace noise and geometric mechanism.
Planning experiment with Gaussian noise and RDP/zCDP.
Failure probability $\delta \approx 10^{-14}/4$

- $\approx 4 \times 10^8$ people
- $10^{-6}$ chance of failure
- Based on $(\epsilon, \delta)$-DP algorithm that returns a random record with probability $10^{-6}$

Moment accountant privacy budget split across 6 levels of geographic hierarchy.

For identity queries:

<table>
<thead>
<tr>
<th>$\epsilon$</th>
<th>Laplace Variance</th>
<th>Gaussian Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>288.0</td>
<td>763.6</td>
</tr>
<tr>
<td>2</td>
<td>72.0</td>
<td>193.9</td>
</tr>
<tr>
<td>3</td>
<td>32.0</td>
<td>87.5</td>
</tr>
<tr>
<td>4</td>
<td>18.0</td>
<td>50.0</td>
</tr>
<tr>
<td>5</td>
<td>11.5</td>
<td>32.4</td>
</tr>
</tbody>
</table>
Gaussian variance is larger than Laplace
But tails are lighter (fewer outliers)
May affect postprocessing steps
Might have better tuned query workload
So experiments are planned (but many other problems need solving)
Thank You