Generating Microdata with Complex Invariants under Differential Privacy

Philip Leclerc, Mathematical Statistician
Center for Enterprise Dissemination-Disclosure Avoidance
United States Census Bureau
philip.leclerc@census.gov

2019 Joint Statistical Meetings

With thanks to the 2020 Disclosure Avoidance System (DAS) development team & our academic partners:

DAS Project Lead:
John Abowd; U.S. Census Bureau & Cornell University

Internal Census Development team:
Robert Ashmead, Simson Garfinkel, Michael Ikeda, Brett Moran, Edward Porter, William Sexton, Pavel Zhuravlev; U.S. Census Bureau

Academic partners:
Michael Hay, Colgate University
Daniel Kifer, Pennsylvania State University (*DAS Scientific Lead*)
Ashwin Machanavajjhala, Duke University
Gerome Miklau, University of Massachusetts Amherst

Outline

DP is a restriction on data publication mechanisms

- DP is a restriction on data publication mechanisms that allows data curators & survey participants to reason rigorously about the degree of privacy risk (risk of breach of confidentiality) incurred due to survey participation
- DP requires probability of outputting any set of final tabulations T cannot depend "very much" on any single input:

$$Pr[\mathbb{M}(X) \in S] \leq e^{\epsilon} Pr[\mathbb{M}(Y) \in S]$$

for all possible neighboring databases X, Y, and possible output subsets S

DP and formally private methods have a number of important properties

- Some notable properties of DP:
  - Enables clear, general proofs bounding privacy risk due to survey participation
  - Requires noise infusion
  - Is a definition, not a mechanism. Many mechanisms are DP
  - Requires considerable expertise when complex large-scale microdata is required as output
- I will use "formal privacy" (FP) to denote related definitions that relax the strength of DP's restrictions, but share its emphasis on provable privacy guarantees against general classes of attackers (including DP itself)
- In practice, formally private methods tend to look & act very much like DP. I am aware of no other methods with general, provable privacy guarantees

Outline

The goal: a formally private Census

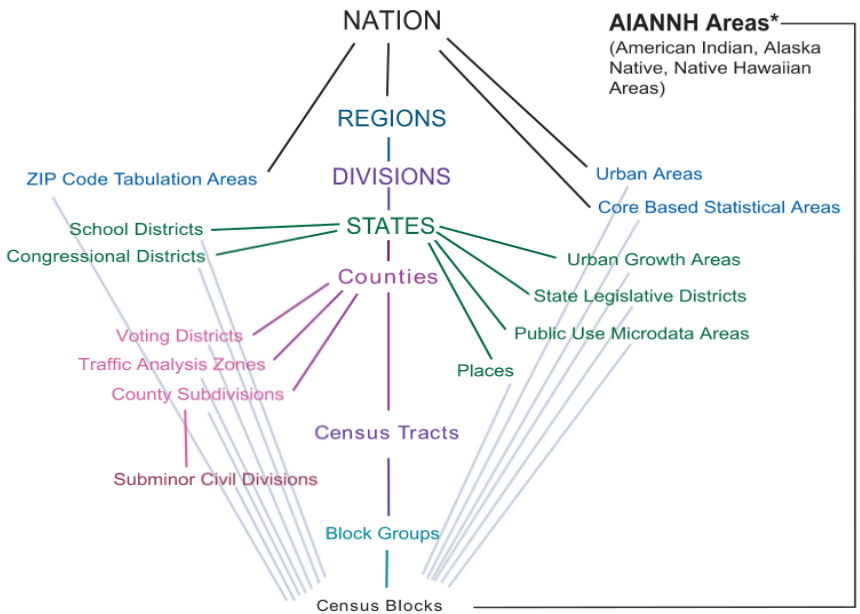
- The 2020 Decennial Census Disclosure Avoidance System (DAS) is the formally private system under development to protect the 2020 Decennial Census
- The DAS expects as input:
  - CEF: Census Edited File, sensitive input data
  - $\mathbb{I}$: invariants, queries with no noise infused
  - W: workload, queries on which we minimize error
- DAS is expected to generate a *Microdata Detail File* (MDF):
  - Define: $\text{MDF} := \text{DAS}(\text{W}, \mathbb{I}(\textit{CEF}), \mathbb{M}(\text{CEF}))$
  - Require: $q(\text{MDF}) = q(\textit{CEF}) \forall q \in \mathbb{I}$
  - Require: $\mathbb{M}$ is ϵ-differentially private
- **Generating good FP microdata is hard, but expected. Today we'll talk about how we're working to achieve that for the Decennial Census.**

The DAS workload is large, complex, & sparse

- Queries in W . . .
  - are defined for geographic units in many geographic levels
  - pertain to two basic record types:
    - Persons
    - Units (Households, Group Quarters Facilities)
  - are organized into 4 major products:
    - PL94+CVAP: $|W_{\text{PL94}}| \approx 7.2$B queries
    - SF1: $|W_{\text{SF1}}| \approx 22$B Person, ≈ 4.5B HH/GQ queries
    - SF2: $|W_{\text{SF2}}| \approx 50B$ queries
    - AIANSF: $|W_{\text{AIANSF}}| \approx 75B$ queries
  - . . . and are required for ≈ 10 other, smaller data products!
- Given $|W|$, we can expect very sparse data
  - $\approx$ 330M person records
  - $\approx$ 125M household records

The DAS workload lives on a geographic lattice

W is organized along a geographic lattice, with increasing sparsity in lower geographic levels:



I refer to levels of this lattice as *geolevels* (e.g., "Blocks", "States"), & units within levels as *geounits* (e.g., "Texas").

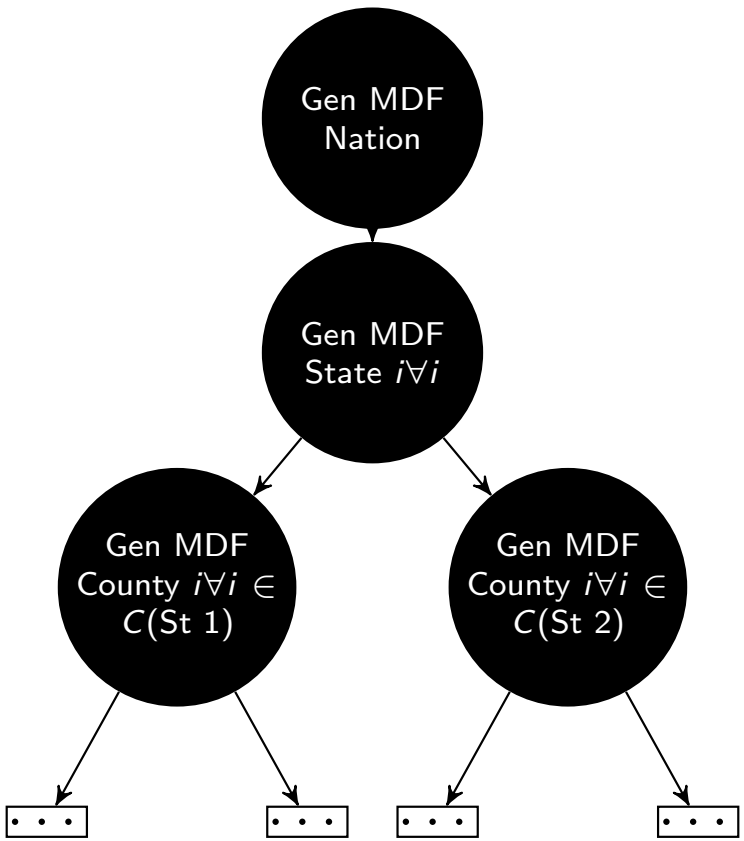
DAS divides work by data product, record type, & geounit

- For each major product D & record type r, we form a schema $\mathbb{S}_{D,r}$. For example,
  - $\mathbb{S}_{\text{PL94,Person}} = \text{VA} \times \text{HHGQ} \times \text{HISP} \times \text{RACE}$
  - With variables defined by:
    - $VA = \{\text{Voting Age, Not Voting Age}\}$
    - $\text{HHGQ} = \{\text{HH}, \text{GQ1}, \ldots, GQ7\}$
    - $\text{HISP} = \{\text{Hispanic, Non-Hispanic}\}$, $\text{RACE} = \{0, \ldots, 2^6 - 2\}$
- For each D, r & geounit g we form a histogram $MDF_{D,r,g} = \mathbb{H}_{D,r,g} \in \mathbb{N}^{|\mathbb{S}|}$
  - Materializing $\mathbb{H}_{D,r,g}$ is expensive: $\approx 2K, 500K, 1M, 10M, 30M, 30M, 85M$ cells per geounit for PL94 Persons, SF1 Households, SF1 Persons, SF2 Persons, SF2 Households, AIANSF Persons, AIANSF Households, resp.
  - But histograms are convenient:
    - Easy to guarantee $\lim_{\epsilon \to \infty} DAS = CEF$
    - Allows generation of microdata consistent with $\mathbb{I}_{D,r,g}$ while simultaneously fitting to all $q \in W_{D,r,g}$

DAS makes $MDF_{D,r,g} = \mathbb{H}_{D,r,g}$ breadth-1st in g

- Follows the "central geopath": Nation, State, County, Tract, Block group, Block
- Top-down movement helps estimate sparsity & controls error for large geounits (vs linear increase in # Census blocks) Divides-and-conquers to control time/RAM requirements
- For each data product, record type D, r:
  - **Phase 1**:
    - For all geolevels & geounits: get DP measurements $\hat{\mathbb{M}} = (\text{HDMM}(W))(CEF)$
    - HDMM is the *High Dimensional Matrix Mechanism* (algorithm for choosing which DP measurements to take)
  - **Phase 2.1**:
    - Compute MDF_{Nation}
    - Consistent* with $\mathbb{I}_{\text{Nation}}$, fitted to $W(\hat{\mathbb{M}}_{\text{Nation}})$
  - Loop: **Phase 2.g**: for each geounit g with MDF_g and children $C(g) \neq \emptyset$, generate $MDF_{g'}, \forall g' \in C(g)$, $\sum_{g' \in C(g)} MDF'_g = MDF_g$

The DAS moves down the central geopath, which expands into a rooted tree



Outline

How do we know the DAS will successfully make microdata?

- DAS is expected to impose invariants in every block-level geounit, but DAS generates microdata at National level, then extends to State level, then to County level, and so on
- Each extension tries to solve a mixed-integer quadratic program (MIQP) to determine good extension of the microdata to next lower level in the central geopath
- How do we know microdata-extension MIQP is always integer-feasible?
- Even if MIQP is integer-feasible, how do we ensure DAS can *find* an integer-valued solution?

DAS tries to solve a MIQP in each non-leaf geounit

$$\arg \min_{H^*_\alpha, \alpha \in c(\gamma)} \sum_{\alpha \in c(\gamma)} \sum_{i \in |\text{rows}(W(\alpha))|} ||W_{i,\star}(\alpha)(H^*_\alpha) - \tilde{\mathcal{M}}(\alpha)_i||_2^2 \quad (1)$$

$$\text{s.t.} \quad (2)$$

$$H^*_\alpha \geq 0 \quad \forall \alpha \in c(\gamma)$$

$$AH^*_\alpha \text{ sign rhs } \forall (A, \text{sign}, \text{rhs}) \in \mathcal{C}(\alpha), \forall \alpha \in c(\gamma) \quad (3)$$

$$\sum_{\alpha \in c(\gamma)} H^*_\alpha = \tilde{H}_\gamma \quad (4)$$

$$H^*_\alpha(s) \in \{0, 1, \dots\} \forall s \in \times_{v \in \mathbb{S}} v, \forall \alpha \in c(\gamma) \quad (5)$$

But MIQP is intractable, so DAS instead solves its continuous relaxation

$$\arg \min_{H^*_\alpha, \alpha \in c(\gamma)} \sum_{\alpha \in c(\gamma)} \sum_{i \in |\text{rows}(W(\alpha))|} ||W_{i,\star}(\alpha)(H^*_\alpha) - \tilde{\mathcal{M}}(\alpha)_i||_2^2 \quad (6)$$

$$\text{s.t.} \quad (7)$$

$$H^*_\alpha \geq 0 \quad \forall \alpha \in c(\gamma)$$

$$AH^*_\alpha \text{ sign rhs } \forall (A, \text{sign}, \text{rhs}) \in \mathcal{C}(\alpha), \forall \alpha \in c(\gamma) \quad (8)$$

$$\sum_{\alpha \in c(\gamma)} H^*_\alpha = \tilde{H}_\gamma \quad (9)$$

$$H^*_\alpha(s) \in \mathbb{R} \forall s \in \times_{v \in \mathbb{S}} v, \forall \alpha \in c(\gamma) \quad (10)$$

DAS then solves an L1 "rounder" problem to get integer-valued solutions

$$\widetilde{H}^0 = \arg \min_{H^\dagger_\alpha, \alpha \in c(\gamma)} \sum_{\alpha \in c(\gamma)} -(H^\dagger_\alpha - \lfloor H^*_\alpha \rfloor) \cdot (H^*_\alpha - \lfloor H^*_\alpha \rfloor) \tag{11}$$

$$\text{s.t. } H^\dagger_\alpha \geq 0 \forall \alpha \in c(\gamma) \text{ (nonnegativity)}$$

$$\sum_s H^\dagger_\alpha[s] = \sum_s H^*_\alpha[s] \forall \alpha \in c(\gamma)$$

$$|H^\dagger_\alpha[s] - H^*_{\alpha|}[s]| \leq 1 \forall \alpha \in c(\gamma), \forall s \in \times_{v \in \mathbb{S}} v$$

$$A H^\dagger_\alpha \text{ sign rhs } \forall (A, \text{sign}, \text{rhs}) \in \mathcal{C}(\alpha) \forall \alpha \in c(\gamma) \tag{12}$$

$$\forall s: \quad H^\dagger_\alpha[s] \in \{0, 1, 2, \dots\} \forall \alpha \in c(\gamma) \tag{13}$$

Outline

Approach # 1: the L2 failsafe & the true data

- It turns out that the DAS can fail in operation! Example:
- The DAS may contain invariants on the number of Group Quarters facilities by type, & on number of Housing Units
- Suppose blocks $\mathcal{B}_1, \mathcal{B}_2$ are the only blocks in Block group BG, with $|\mathcal{B}_1| = |\mathcal{B}_2| = 100$, 1 GQ of types A and B in $\mathcal{B}_1$, and 1 GQ of type C in $\mathcal{B}_2$
- When inferring microdata in BG, this implies the obvious constraints:
  1. $|\text{BG}| = 200$
  2. $|\text{BG}_{GQ_A}| \geq 1$
  3. $|\text{BG}_{GQ_B}| \geq 1$
  4. $|\text{BG}_{GQ_C}| \geq 1$

Approach # 1: the L2 failsafe & the true data

- But suppose we had inferred $|\text{BG}_{GQ_A}| = |\text{BG}_{GQ_B}| = 1, |\text{BG}_{GQ_C}| = 198$. We then don't have enough GQ_A, GQ_B people to "fill in" $\mathcal{B}_1$!
- In these cases, our last line of defense is a *failsafe post-processor*. It converts the L2 problem's hierarchical consistency constraint into an objective function penalty
- The resulting variant of QP (6)-(10) is then feasible: *the true data satisfies it*
- But what about the "rounder" IP/LP, (11)-(13)?

Approach # 1: the L1 failsafe & TUM

- In general, even integer L2/QP feasibility does not imply IP/LP/L1 "rounder" feasibility
- To fix this, *total unimodularity* (TUM) is useful: matrix A is TUM iff every subdeterminant of A is in $\{-1, 0, 1\}$
- Roughly speaking, TUM matrices characterize polyhedra with integer "corner points"
- If the left-hand-side matrix in LP (11)-(13) is TUM and the QP is continuous-feasible, it follows that the LP rounder is integer-feasible
- Moreover, standard mathematical programming algorithms can then be used to solve (11)-(13) in polynomial time for integer solution

Approach # 2: implied constraints & cutting planes

- Although the failsafe provably works, it also harms statistical utility / accuracy, because it sacrifices hierarchical consistency
- Moreover, TUM is fragile when expanding invariants set, so the failsafe may fail to work for more complex invariant sets
- To improve accuracy & flexibility of the DAS, we also investigate a broader approach:
  1. In each node, DAS should compute over a non-empty subset of integer hull of the projection of all block-level solutions
  2. Infeasibilities in DAS stem from "missing" inequalities: present in the projection, but not in DAS

Approach # 2: implied constraints & cutting planes

- Note the primary feature of the "GQs" example problem: non-obvious block-level information was not properly incorporated into optimization problems at higher geolevels
- We have identified constraints sufficient to capture this missing information (next slide)
- No do not yet have mathematical proof of this set's sufficiency, but significant empirical evidence supports it

Approach # 2: implied constraints & cutting planes

- Empirically, the following class of inequalities appears to be sufficient to ensure integer feasibility in all intermediate DAS sub-problems, without invoking the failsafe:

$$LB(B,S) = \begin{cases} T_B & S \supseteq P(B) \\ \sum_{i \in S} f_{B,i} & o.w. \end{cases}$$

$$UB(B,S) = \begin{cases} 0 & S \cap P(B) = \emptyset \\ T_B - \sum_{i \notin S} f_{B,i} & o.w. \end{cases}$$

$$LB(N,S) = \sum_{B \in leaves(N)} LB(B,S) \tag{14}$$

$$UB(N,S) = \sum_{B \in leaves(N)} UB(B,S) \tag{15}$$

- But this class is *exponentially large*!

Approach # 2: implied constraints & cutting planes

- Facing exponentially many inequalities motivates us to consider *cutting planes*
- Cutting-plane techniques incrementally add inequalities as needed to a relaxation of some target optimization problem
- Cutting planes can be useful when a target optimization problem requires intractably many inequalities to describe

Approach # 2: a cut-plane generator

$$\text{argmin}_{b_i, b_R} \left(\sum_{i=0}^{|HHGQ|-1} x^*_{N,i} b_i \right) - \sum_{R \in \mathcal{R}} \left(T_R b_R + (1 - b_R) \sum_{i \in P(R)} f_{R,i} b_i \right) \tag{16}$$

subject to

$$b_R \leq \frac{1}{|P(R)|} \sum_{i \in P(R)} b_i \forall R \in \mathcal{R} \tag{17}$$

$$b_i, b_R \in \{0, 1\} \forall i, R \tag{18}$$

Approach # 3: finding feasibility in network flows

- This approach is relatively new, so I don't want to say too much about it right now!
- Briefly, there seem to be natural ways to describe feasibility the "non-obvious missing inequalities" in our problem in terms of network flows
- This approach relates in some natural ways to earlier approaches; notably, the *number of Group Quarters facilities combinations with non-zero counts in some block* again figures prominently in complexity
- We are exploring efficient ways to incorporate and test this approach

Contact Information

Thanks for listening! If you have follow-up questions, I can be reached at:

Email: philip.leclerc@census.gov