Leveraging Vertical Public-Private Split for Improved Synthetic Data Generation

Samuel Maddock Shripad Gade Graham Cormode Will Bullock

Email: {smaddock, shripadgade, gcormode, bullock}@meta.com

1. Differentially Private Synthetic Data

Goal: Produce "artificial" data with the underlying statistical properties of source data

Applications: Synthetic Data allows access to realistic data without compromising privacy and enables downstream tasks e.g., training ML models, data analytics.

- **Differentially Private Synthetic Data Generation (DP-SDG)** algorithms involve adding carefully calibrated noise in the training / modeling process to provide provable **Differential Privacy (DP)** guarantees.
- DP-SGD algorithms typically assume that **all source data is private**.

Horizontal & Vertical Partitioning in Data: Real-world datasets naturally contain a mix of public and private information.

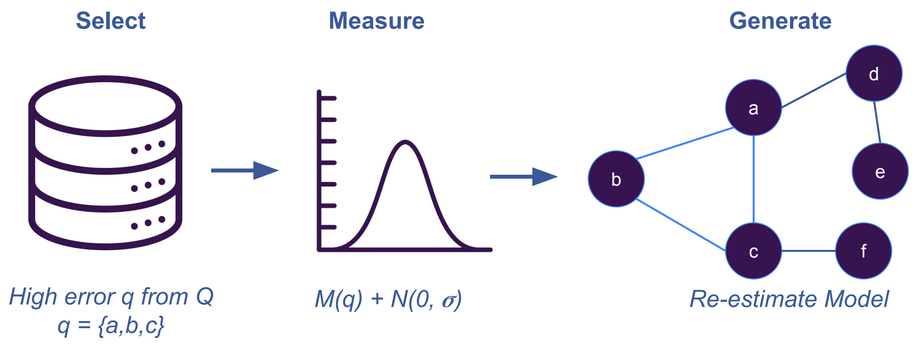
- **Horizontal:** A fraction of the rows are considered public.
- **Vertical:** A fraction of the columns are considered public.

This paper: How do we leverage vertical public-private partitioned data to improve synthetic data generation?

2. "Select-Measure-Generate" Approach

Tabular DP-SDG methods follow **"Select-Measure-Generate"** approach.
At each iteration $t = 1, \ldots, T$:

1. **Select** query q in workload with worst error (noisily) with utility scores $s(q)$
2. **Measure** chosen query q under calibrated Gaussian noise
3. **Generate** synthetic data and update model to learn (noisy) measured queries



Existing methods instantiate this framework, changing the synthetic data model used:

- **MWEM:** Models the synthetic distribution directly, suffers from scalability issues.
- **AIM:** Learns a graphical model via Private-PGM, obtaining SOTA utility.
- **GEM:** Train a generator neural network on noisy measurements.

3. Prior Work: Horizontal Public-assisted

Horizontal Partitioning: Prior work assumes publicly available subset of *rows* which is leveraged to improve synthetic data utility. Existing approaches include:

- **Pre-training approaches:** Initializes synthetic data model using the public dataset via pretraining. Methods include **PMWpub** which pretrains MWEM, and **GEMpub** which pretrains a generator neural network.
- **In-training approaches:** Directly use public data during private training. **JAM-PGM** modifies "Select" step of AIM to choose either a public or private marginal.

4. Our Work: Vertical Public-Assisted

Vertical Partitioning: Assumes subset of *columns* are considered public while rest are private. This hasn't been well-studied before.

Framework to adapt existing public-assisted methods to the vertical partitioned setting. This involves modifying pre-training and measurement steps in existing algorithms to get variants: **vPMWpub, vGEMpub, vJAM-PGM**

Algorithm 1 Vertical Public-assisted Adaptive Measurements (vPAM)

Input: Private dataset D_{priv}, public dataset D_{pub}, workload of queries W, training steps T, privacy parameters (ε, δ)
Output: Synthetic data $\hat{D}$

Pre-process the workload $W^* := \text{PROCESS-WORKLOAD}(W)$
$\theta_0 := \text{MODEL-INIT}(D_{\text{pub}})$, $\hat{D}_0 \sim \theta_0$
for $t = 0, \ldots, T-1$ **do**
  Select: via the Exponential mechanism $q_t \in W^*$ using $\text{SCORE}(q; D_{\text{priv}}, D_{\text{pub}}, \hat{D}_t)$
  Measure: selected marginal query q_{t+1} i.e., $\tilde{M}_{t+1} := \text{MEASURE}(q_{t+1}; D_{\text{priv}}, D_{\text{pub}}, \sigma^2)$
  Update: synthetic model $\theta_{t+1} := \text{TRAIN-MODEL}(\theta_t, \{\tilde{M}_1, \ldots, \tilde{M}_{t+1}\})$
  Generate: $\hat{D}_{t+1} \sim \theta_{t+1}$
end for
Output $\hat{D} \sim f(\{\theta_t\}_{t=1}^T)$

5. Our Work: Conditional Generation

Adaptations of existing methods encounter issues:

- **Pretraining approaches** are brittle because private training can undo much of the information learnt from pretraining.
- **In-training approaches** i.e., JAM-PGM wastes some privacy budget deciding whether to select public or private marginals.

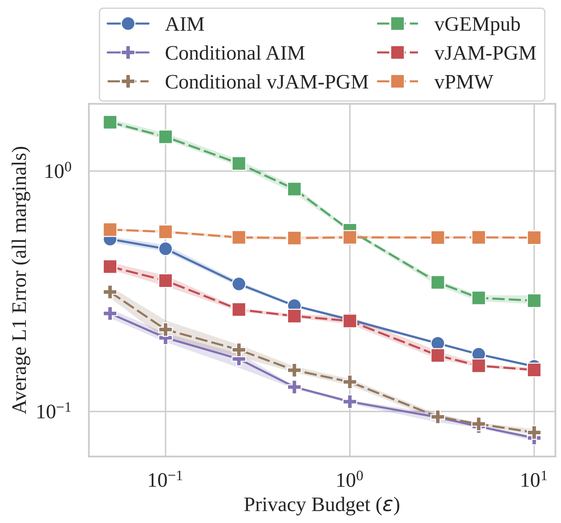
In the vertical setting we have direct access to public columns and can use this to improve the sampling process → **Conditional AIM**, variant of **AIM** that modifies Private-PGM to use the raw data for public columns (i.e., exact marginals)

6. Empirical Setup

Train methods on a workload of 3-way marginals and measure **average L1 error**. Experiments on Adult and Census-Income datasets, comparing fully private **AIM** (baseline) with adapted algorithms from our framework:

.	vPMWPub	vGEMPub	vJAM-PGM	Conditional AIM
Pre-training	✓	✓	×	×
Select	Unmodified (MWEM)	Unmodified (MWEM)	Modified AIM (choose private or public marginal)	Modified AIM (choose marginal w/ at least 1 private column)
Measure	Unmodified (Gaussian)	Unmodified (Gaussian)	Gaussian (private) or zero-noise (public)	Unmodified (Gaussian)
Generate	Multiplicative weights	Generator network	Private-PGM w/ uniform weights	Private-PGM w/ conditional sampling

7. Results: Initial Comparison



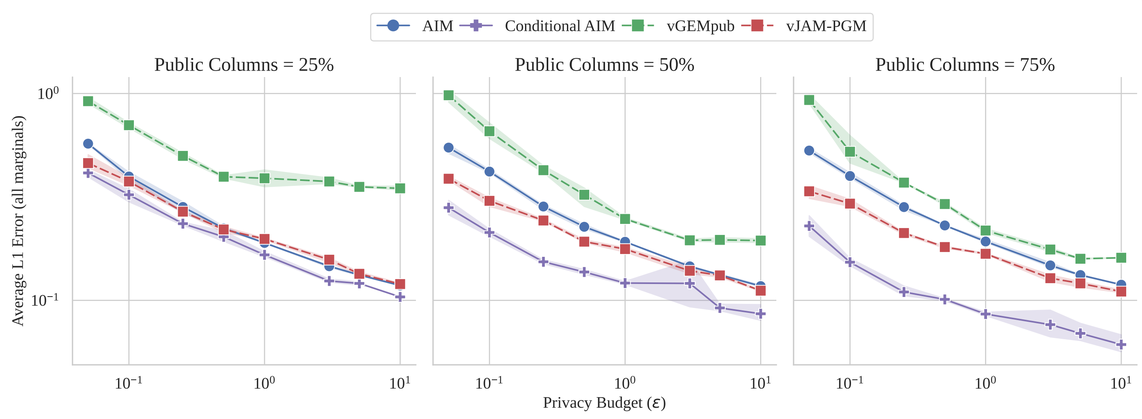
Initial comparison on Adult (reduced, 6/8 columns taken as public)

vGEMpub and **vPMW** underperform showing pretraining approaches are not useful in vertical setting.

vJAM-PGM shows improved utility over AIM when privacy budget is low.

Conditional AIM outperforms all methods highlighting the effectiveness of conditional generation.

8. Results: Varying Public-Private Splits



Study effects of different proportions of public columns:

- Minimal utility gains over private **AIM** when public columns are $< 50\%$
- Substantial improvements with **Conditional AIM** as the public columns increase
- **vJAM-PGM** is the nearest competitor but only performs well in high-privacy settings

9. Takeaways & Looking Forward

Scalability: Conditional AIM achieves best utility but can require an intractable amount of memory if conditioning on a large number of public columns, this is inherent to Private-PGM.

Utility: Vertical public-assisted methods struggle to remain competitive against (fully) private AIM. Pre-training on public data has little effect on overall utility.

Research Directions: Focus the design of future vertical public-assisted algorithms away from pre-training and towards direct usage of public data during training. Improving the scalability and utility of conditional methods, by improving elimination order and efficient-conditioning for Private-PGM or adapting these methods to generator networks in GEM.

9. Takeaways & Looking Forward