Differentially Private Data Synthesis: State of the Art and Challenges

Presenter: Ninghui Li

Purdue University

2022.06.02 @ ASIACCS

Overview of Differential Privacy

Differential Privacy [Dwork et al. 2006]

- Definition: Mechanism *A* satisfies ε-Differential Privacy if and only if
  - for any neighboring datasets D and D'
  - and any possible transcript $t \in$ Range(A),

$$\Pr[A(D) = t] \leq e^{\epsilon} \Pr[A(D') = t]$$

  - For relational datasets, typically, datasets are said to be neighboring if they differ by a single record.

Why Does Differential Privacy Make Sense?

- "Privacy as Secrecy" (i.e., hiding information) is impossible if one wants to share data at all.
  - Consider the following example: Assume that smoking causes lung cancer is not yet known, and an organization conducted a study that demonstrates this connection.
  - A smoker Carl was not involved in the study, but complains that publishing the result of this study affects his privacy, because others would learn new information about him, namely he has a higher chance of getting lung cancer, and as a result he may suffer damages.
- Differential Privacy attempts to simulate "privacy as opting out": The most one can do to protect one's privacy is to take one's data out of the dataset.

Using an ϵ-DP Definition is Good Enough When

1. For each dataset D, and each individual X whose data is in D, there exists an dataset D' such that
   - D and D' are neighboring
   - All data that belong to X in D have been removed or overwritten in D'
   - That is, we can say that D' is "an ideal world of privacy" for individual X
2. The privacy parameter ϵ is suitable for the setting

There are many applications of ϵ-DP where one cannot automatically assume that using ϵ-DP provides strong privacy protection.

Chapter 3 of Li et al.: Differential Privacy: From Theory to Practice. Morgan & Claypool Publishers 2016.

Nice Properties of DP

- Post-processing Invariance
  - If A_1 satisfies ε_1-DP, then $A_2(A_1(\cdot))$ always satisfies ε_1-DP
- Sequential Composability
  - If A_1 satisfies ε_1-DP, and A_2 satisfies ε_2-DP, then outputting both A_1 and A_2 satisfies $(\varepsilon_1+\varepsilon_2)$-DP.
  - ε Is known as the privacy budget.
  - Enables modular design of DP algorithms
- Parallel Composability
  - If D is partitioned into two parts, applying A_1 and A_2 on the two parts satisfy $(\max(\varepsilon_1,\varepsilon_2))$-DP

Better Sequential Composition Properties

- High-level Message: When composing T steps (where T is dozens or larger), the total privacy cost grows (roughly) proportional to $T^{1/2}$
  - For privacy, we want the two distributions $A(D)$ and $A(D')$ to be close.
  - ϵ-DP bounds worst-case of privacy loss on any output
  - The notion of (ϵ,δ)-DP relaxes ϵ-DP and is generally acceptable when δ is small
  - Other generalizations of DP, such as zero-Concentrated DP and Renyi DP, help more accurately analyze privacy loss in composition of mechanisms
- Increase the size of dataset by a factor of *C* means that the number of queries that can be answered with similar accuracy grows by a factor of C^2.

Abadi et al.: Deep Learning with Differential Privacy. CCS 2016
Bun & Steinke: Concentrated Differential Privacy: Simplifications, Extensions, and Lower Bounds. TCC 2016.
Ilya Mironov: Rényi Differential Privacy, 2017 CSF.

Differential Privacy: Basic Mechanisms

- Laplace Mechanism:

$$A_f(D) = f(D) + Lap(\frac{\Delta_f}{\varepsilon})$$

- Exponential Mechanism: Sample an answer with probability determined by the quality of the answer.

Answering Queries Under DP

- Easy queries: their answers do not change much when one record changes
  - E.g., counting number of records that satisfy certain conditions.
- Hard queries: their answers are easily affected by one record change
  - E.g., max / min
- Usually easy to answer, but worst-case exist: their answers typically change very little when one record changes, but could change a lot in pathological cases
  - E.g., median
  - One can come up with mechanisms that work well in practice without meaningful proven bound of utility

From Relational Data to Contingency Table

Gender	Age	Income
Female	31	150k
Male	28	100k
Male	30	110k
Male	45	200k
Female	19	50k
Male	24	40k

Gender Male: 1 Female: 0	Age Larger than 25: 1 Otherwise : 0	Income More than 100k: 1 Otherwise: 0	Count
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	3

From Contingency Table to Marginal Tables

Gen	Age	Inc	Cnt
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	3

Gen	Age	Cnt
0	0	1
0	1	1
1	0	1
1	1	3

Gen	Cnt
0	2
1	4

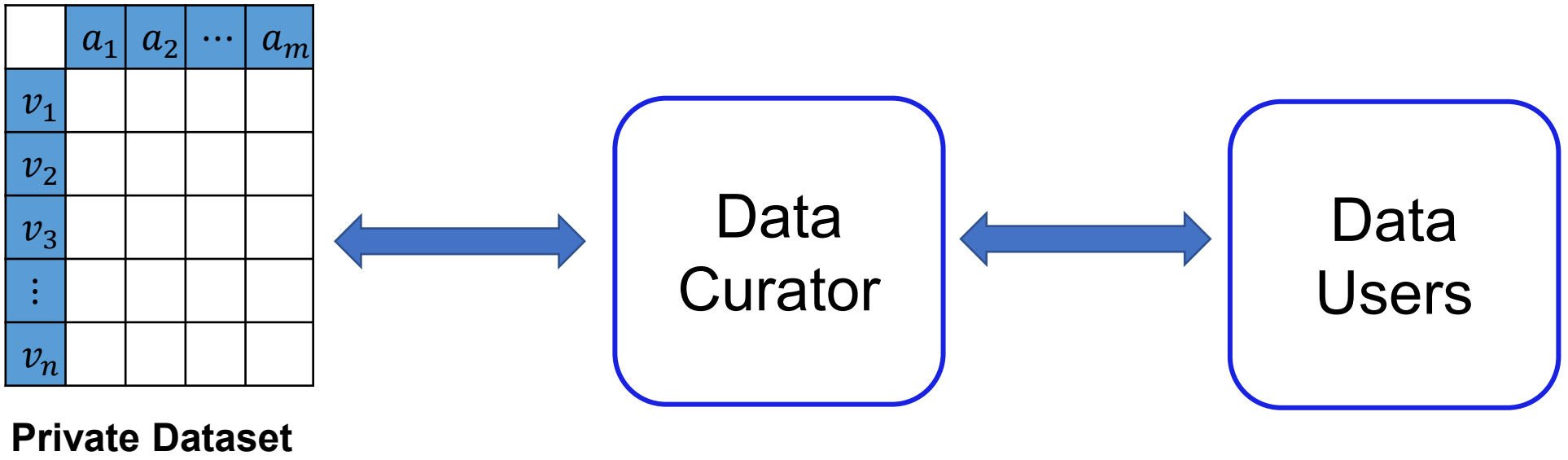
Marginal table queries can be answered with the same privacy cost as each counting query. They are arguably the most privacy-efficient way to extract information from input dataset.

Settings of DP

Four Settings for Using DP

- Local setting (aka Local DP)
  - Do not trust server, each user must perturb the data before sending to server.
- Single workload
  - E.g., learning a classifier, finding frequent itemsets, etc.
- Interactive setting
  - A data curator/server answers queries as they come, not knowing what the future queries are
- Non-interactive
  - Publishing a summary of data that can be used to answer future queries
  - Publishing synthetic datasets

Interactive Setting



- A trusted data curator has access to private dataset
- The data curator receive queries from data users
- The data curator answers the queries using private dataset while satisfying DP
- The data curator does not know what queries will be asked in future

Interactive Setting: Motivations and Limitations

- Motivation
  - Lowerbound results on reconstruction attacks demonstrate that answering a linear number of counting queries accurately (with error $o(N^{1/2})$) destroys privacy
  - Hence the goal is to answer a sublinear number of queries accurately
- Limitation
  - Answering each query consumes some privacy budget
  - After answering a pre-determined number of queries, one exhausts the privacy budget, and cannot answer any later query
  - Situation exacerbated when dealing with multiple data users

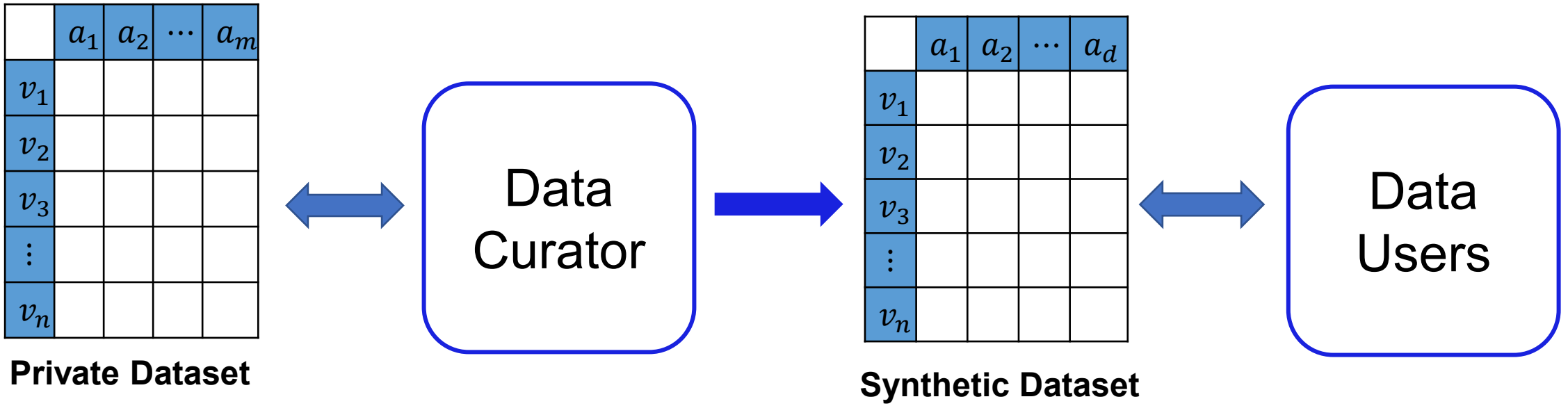
Basic Reconstruction Attacks:

- Section 8.1 of Dwork and Roth: The Algorithmic Foundations of Differential Privacy.
- Dinur and Nissim, Revealing Information while Preserving Privacy, PODS 2003.

Getting Around of Reconstruction Attacks

1. Answers some queries with $\Theta(N^{1/2})$ accuracy is acceptable
   - E.g., this level of accuracy is what can be achieved under LDP
2. Not all queries need to be answered accurately in practice
   - Decide which queries will be answered accurately
3. No need to guarantee accurate answers to all queries for all possible input dataset; it suffices to
   1. Answer some queries with guaranteed accuracy
   2. Answer some other queries with high accuracy most of the time (depending on input datasets), but errors can be large in pathological cases

Non-interactive: Synthetic Data Generation



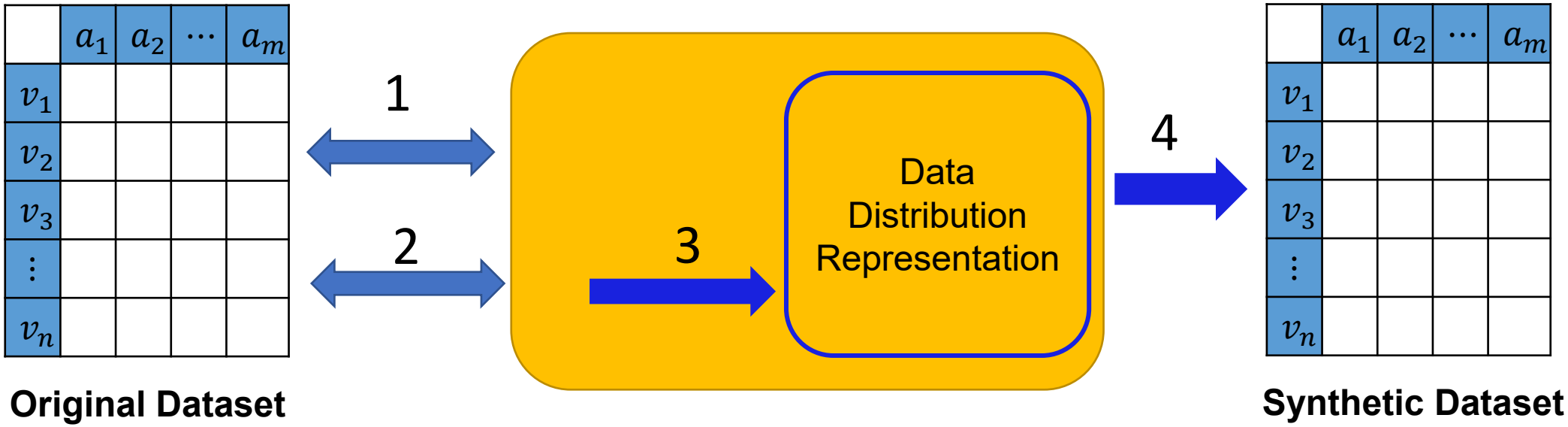
- Develop an algorithm that
  - Takes a private dataset as input
  - Generates one or more synthetic datasets that are “similar” to the input dataset, where similar means
    - Provide similar answers on multi-dimensional range queries
    - Certain tasks can be performed with similar accuracy

Need for Private Data Synthesis

- US Census Bureau applies DP in 2020 census
- US NIST ran two DP synthetic data challenges
  - 2018 Differential Privacy Synthetic Data Challenge
  - 2020 Differential Privacy Temporal Map Challenge
- Enable existing data analysis with little change even under DP

Approaches to Private Data Synthesis

Private Synthesis Framework

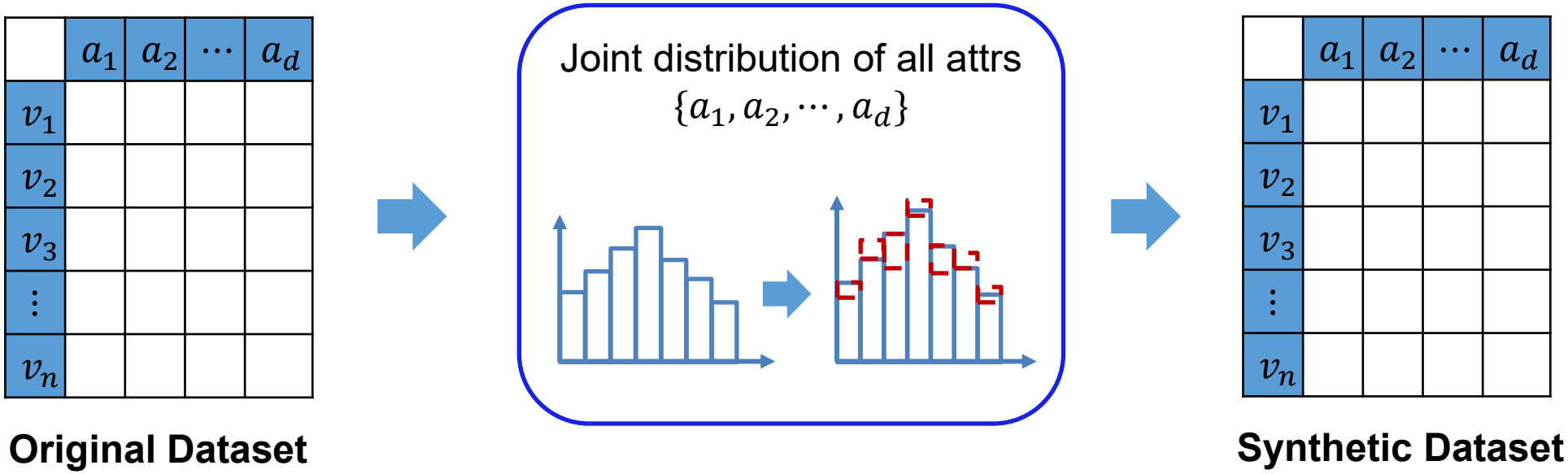


1. Choose appropriate queries (possibly depending on the dataset)
2. Obtain query answers while satisfying DP
3. Construct data representation from query answers
4. Synthesize dataset from the representation

Types of Data Distribution Representation

- Full domain probability density
- Probabilistic Graphical Model
- Neural Network Generative Model
- A set of consistent low-dimensional marginals

Using Full Domain Distribution



Original Dataset

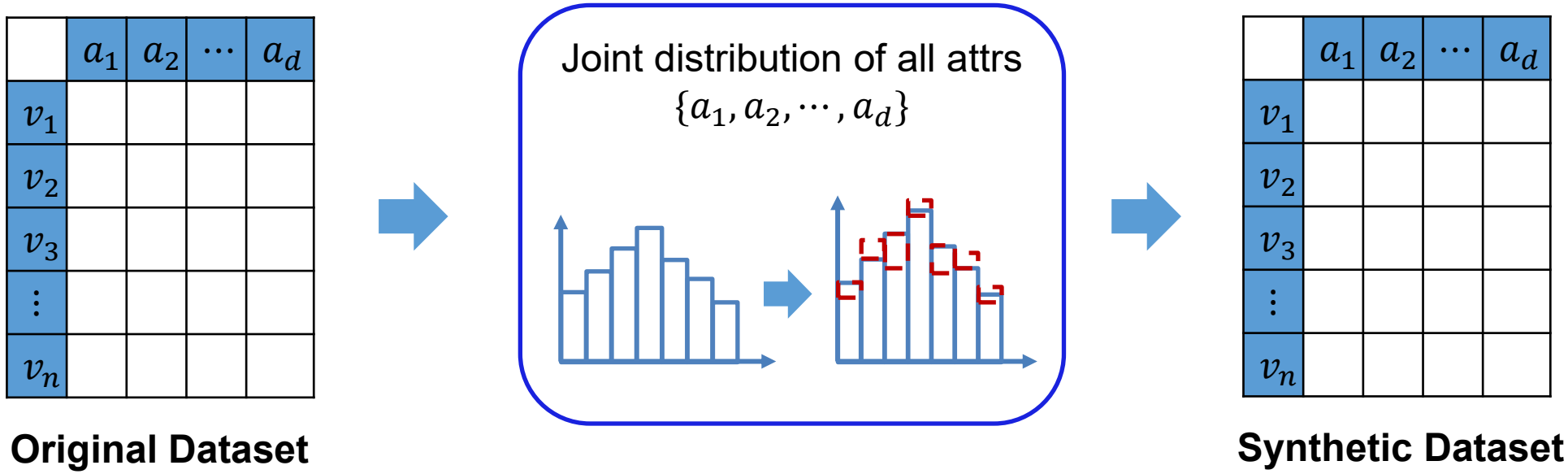
Synthetic Dataset

MWEM: An Example Using Full Domain Distribution

- Multiplicative Weights update rule with the Exponential Mechanism
  - Assumes as input a set of queries one wants to answer accurately
  - Maintains a probability distribution over the whole domain, initialized to be all uniform
  - Repeat a number of times
    - Uses Exponential Mechanism to choose a query that has the largest error given current distribution
    - Obtain a noisy answer to the chosen query
    - Update the distribution using the query answers answer (using the multiplicative update rule)

Hardt, Ligett, McSherry: A Simple and Practical Algorithm for Differentially Private Data Release. NIPS 2012.

Using Full Domain Distribution: Limitations



❖ **When the number of attributes is large, the domain of joint distribution is large, leading to prohibitive computational cost.**

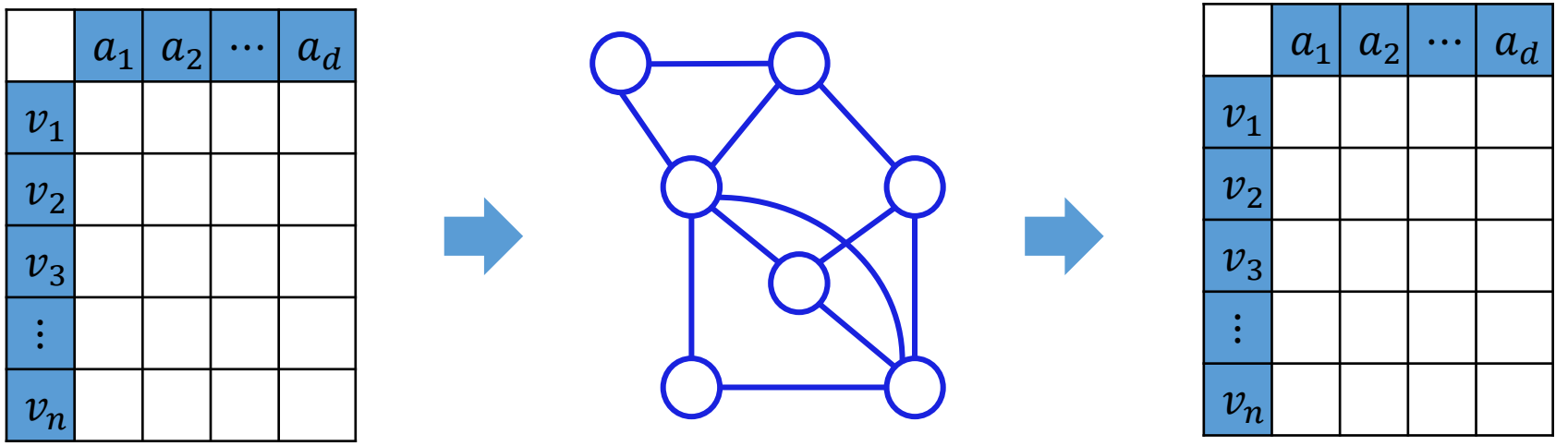
Neural Network Generative Model

Original Dataset **Synthetic Dataset**

- Using Neural Network to encode data distribution, and GAN to train the network
- Does not perform well empirically for tabular data, possibly because
  - Queries do not use privacy budget efficiently.
  - Cannot directly select what information is preserved.
- CTGAN combines one-dimensional marginals with GAN

Xu et al.: Modeling tabular data using conditional GAN. NeurIPS 2019.

Graphical Model



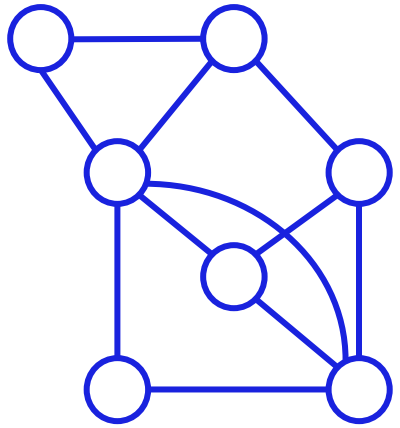
- Uses a Probabilistic Graphical Model (e.g., Bayesian Network or Markov Random Fields) as representation.
- Learn the PGM structure and parameters (through marginal tables)
- Sample the model to generate synthetic data

Zhang et al.: PrivBayes: Private data release via Bayesian networks. SIGMOD 2014.
McKenna, Sheldon, and Miklau: Graphical model based estimation and inference for differential privacy. ICML 2019.

Graphical Model: Some Limitations

	a_1	a_2	$\cdots$	a_d
v_1				
v_2				
v_3				
$\vdots$				
v_n				

Original Dataset



	a_1	a_2	$\cdots$	a_d
v_1				
v_2				
v_3				
$\vdots$				
v_n				

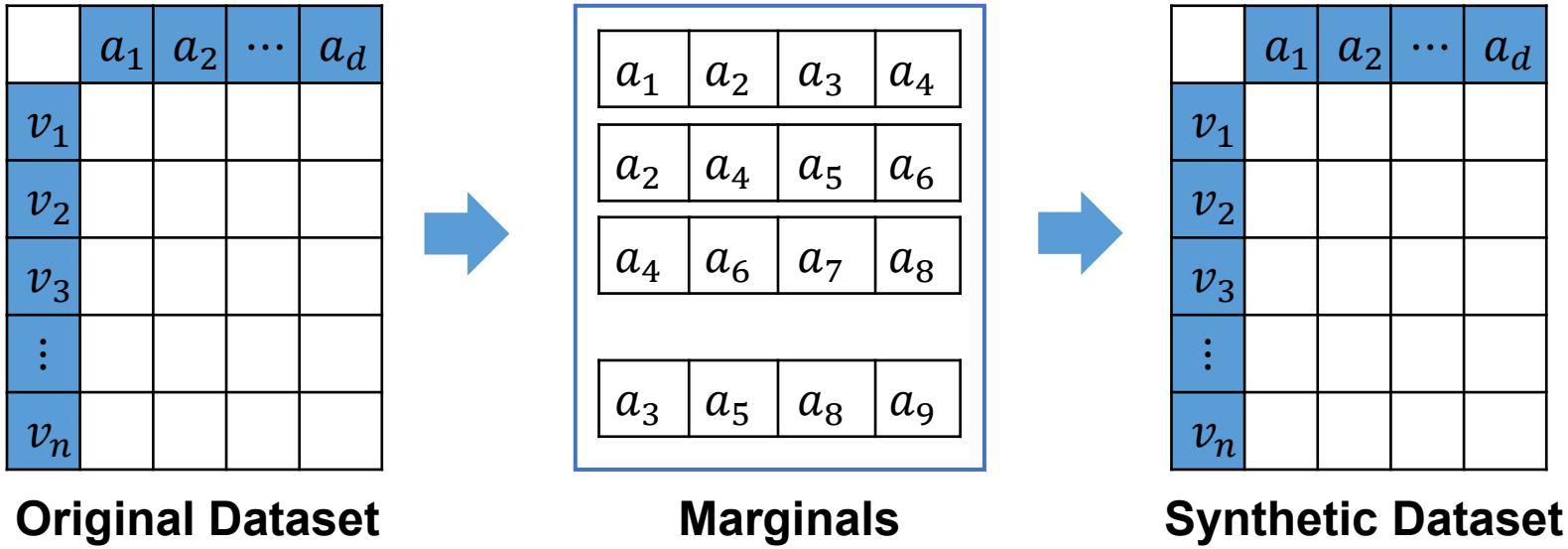
Synthetic Dataset

- **Bayesian Network [1]: Can only exploit d-1 marginals, losing many correlation information.**
- **Markov Random Field [2]: Some cliques can be very large when the number of marginals is large, leading to high storage cost.**

Zhang et al.: PrivBayes: Private data release via Bayesian networks. SIGMOD 2014.
McKenna, Sheldon, and Miklau: Graphical model based estimation and inference for differential privacy. ICML 2019.

PrivSyn: Our Approach

PrivSyn: Our Approach

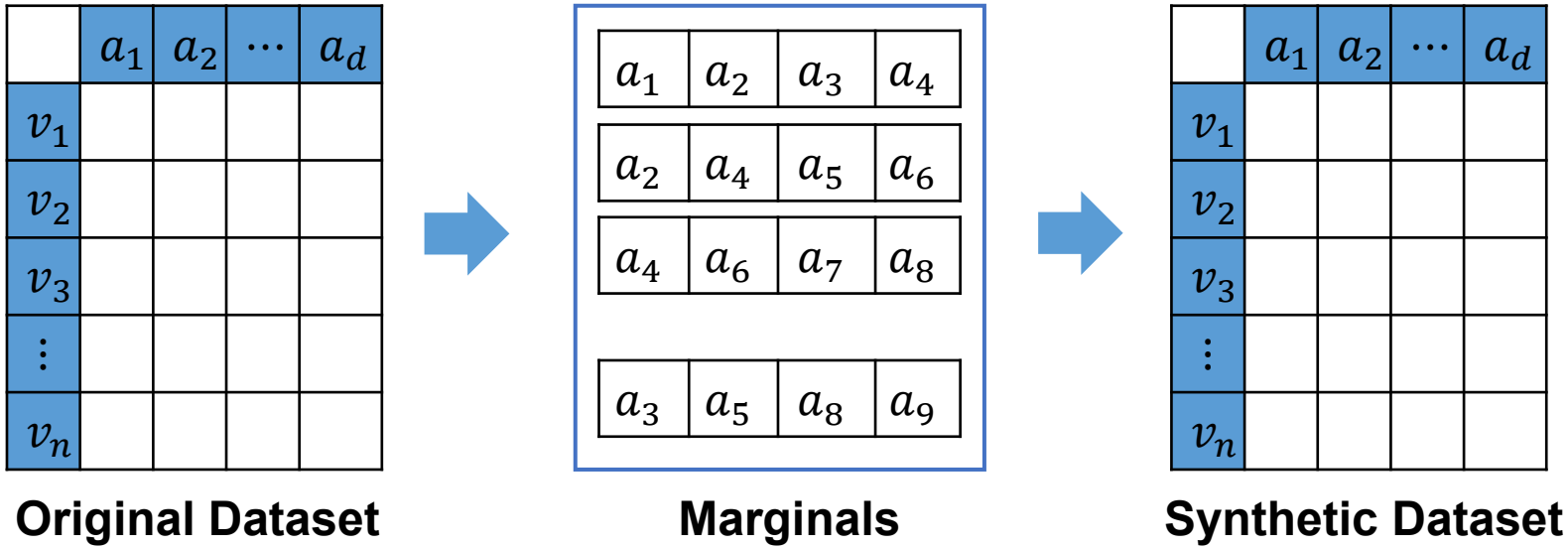


Original Dataset **Marginals** **Synthetic Dataset**

- **Use a set of consistent low-dimensional marginals as representation**
- **Synthesize data directly from marginals**
- **Can be viewed as a non-parametric method compared to PGM**

Qardaji, Yang, and Li: PriView: practical differentially private release of marginal contingency tables. SIGMOD 2014.
Zhang et al.: PrivSyn: Differentially Private Data Synthesis. USENIX Security 2021.

PrivSyn: Our Approach



Original Dataset **Marginals** **Synthetic Dataset**

- **Challenge 1: How to choose a set of marginals that capture as much as correlation information and avoid excessive noise.**
- **Challenge 2: How to generate a synthetic dataset from the selected marginals.**

Marginal Selection: DenseMarg

❑ Example

- ❖ Attributes: a b c d
- ❖ All two-way marginals: (a, b), (a, c), (a, d), (b, c), (b, d), (c, d)

Noise Error

- ❖ If a two-way marginal is chosen
  - Add **Gaussian noise** to obtain the marginals
  - Proportional to the number of cells

Dependency Error

- ❖ If a two-way marginal is **not** chosen
  - Mutual information (high sensitivity)
  - **InDif** (low sensitivity)

Optimization Problem Formulation:

$$\text{minimize} \sum_{i=1}^{m} [\psi_i x_i + \phi_i (1 - x_i)]$$

$$\text{subject to } x_i \in \{0, 1\}$$

Marginal Selection: DenseMarg

Optimization Problem Formulation:

$$\text{minimize} \sum_{i=1}^{m} [\psi_i x_i + \phi_i (1 - x_i)]$$

$$\text{subject to } x_i \in \{0, 1\}$$

Greedy algorithm to solve the optimization problem

Algorithm 1: Marginal Selection Algorithm

Input: Number of pairs m, privacy budget ρ, dependency error $\langle \phi_i \rangle$, marginal size $\langle c_i \rangle$;
Output: Selected marginal set X;

$X \leftarrow \varnothing$; $t \leftarrow 0$; $E_0 \leftarrow \sum_{i \in \bar{X}} \phi_i$;
while *True* **do**
  foreach *marginal* $i \in \bar{X}$ **do**
    Allocate ρ to marginals $j \in X \cup \{i\}$;
    $E_t(i) = \sum_{j \in X \cup \{i\}} c_j \sqrt{\frac{1}{\pi \rho_j}} + \sum_{j \in \bar{X} \setminus \{i\}} \phi_j$;
  $\ell \leftarrow \arg\min_{i \in \bar{X}} E_t(i)$;
  $E_t \leftarrow E_t(\ell)$;
  if $E_t \geq E_{t-1}$ **then**
    Break
  $X \leftarrow X \cup \{l\}$;
  $t \leftarrow t + 1$;

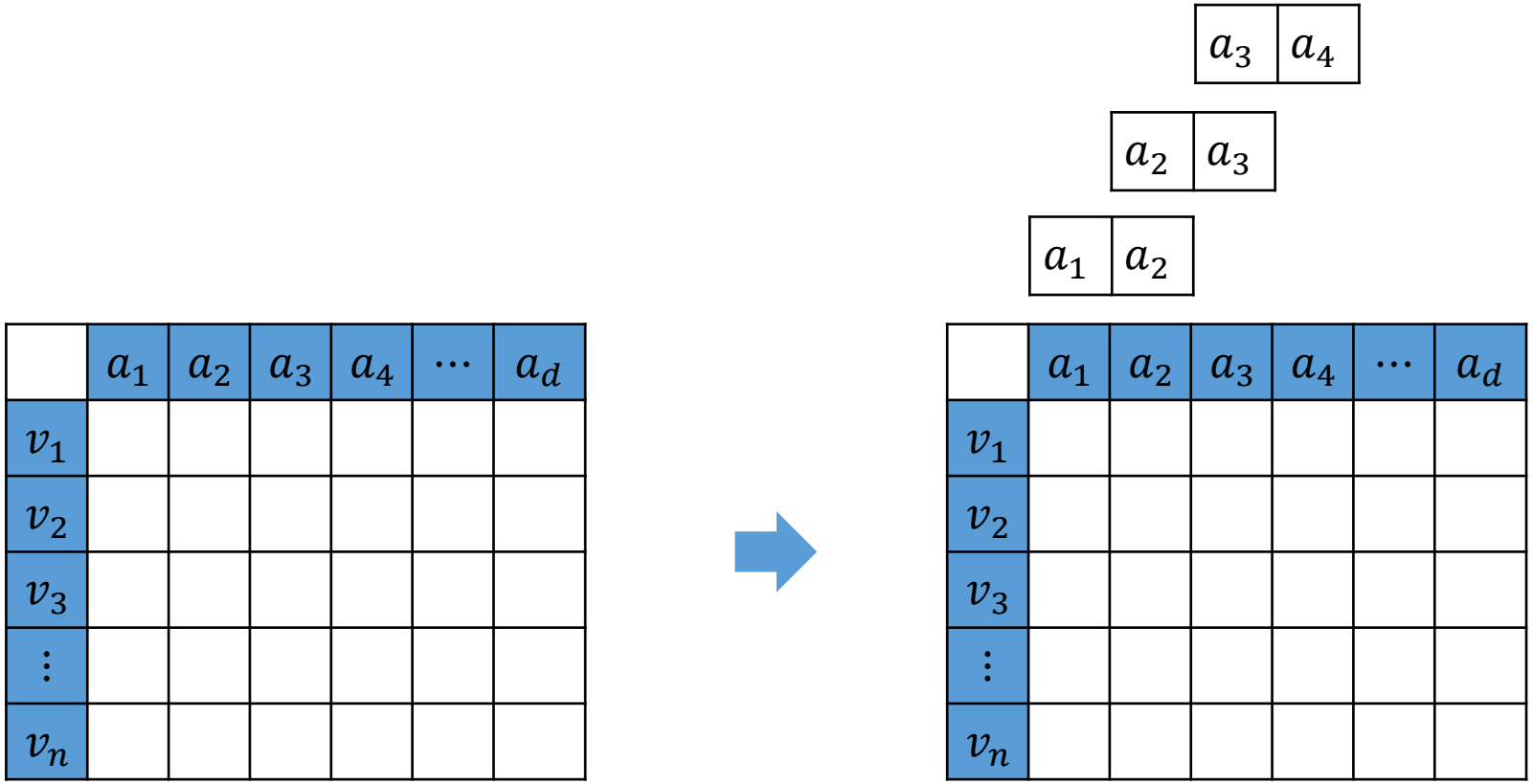
Combine small two-way marginals to larger marginals

Algorithm 2: Marginal Combine Algorithm

Input: Selected pairwise marginals X, threshold γ
Output: Combined marginals $\mathcal{X}$

Convert X to a set of pairs of attributes;
Construct graph $\mathcal{G}$ from the pairs;
$S \leftarrow \varnothing$; $\mathcal{X} \leftarrow \varnothing$
foreach *clique size* s *from* m *to* 3 **do**
  $C_s \leftarrow$ cliques of size s in $\mathcal{G}$
  foreach *clique* $c \in C_s$ **do**
    if $|c \cap S| \leq 2$ *and domain size of* $c \leq \gamma$ **then**
      Append c to $\mathcal{X}$
      Append the attributes of c to S

Dataset Generation: GUM



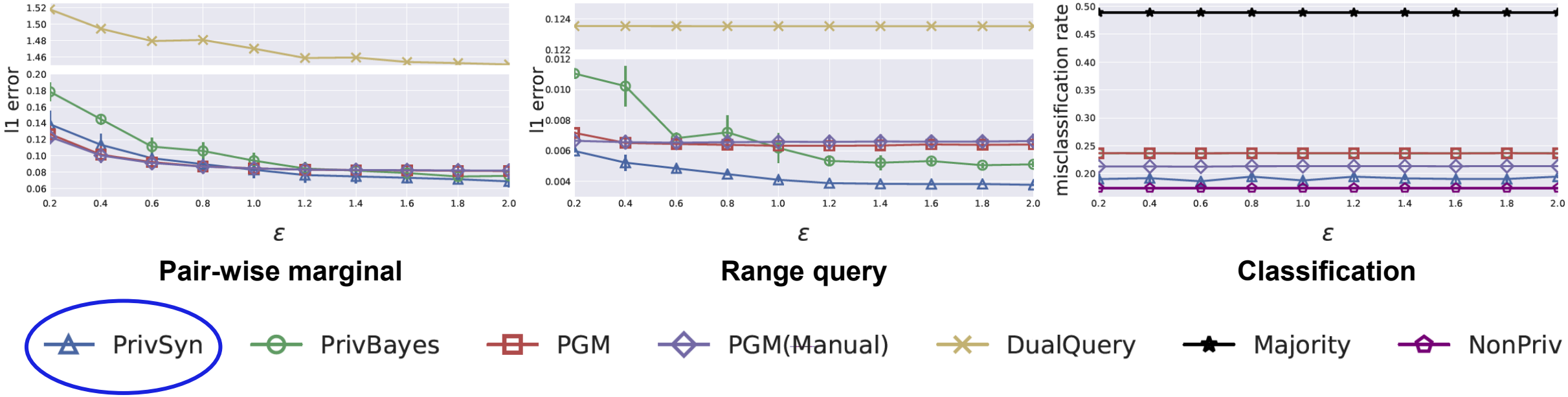
Step I: Randomly generate dataset.

Step II: Iteratively using all marginals to update the dataset.

Experimental Setup

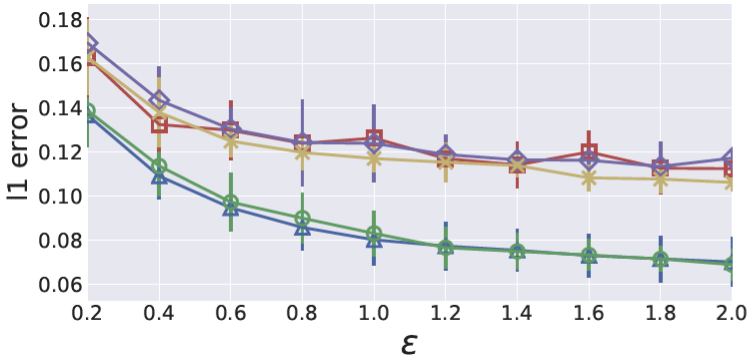
- Tasks and Metrics
  - Pair-wise marginals (Average L1 error)
  - Range query (Average L1 error)
  - Classification (Misclassification rate)
- Competitors
  - Bayesian network (PrivBayes)
  - Markov random field (PGM)
  - Game-based method (DualQuery)

End-to-end Comparison

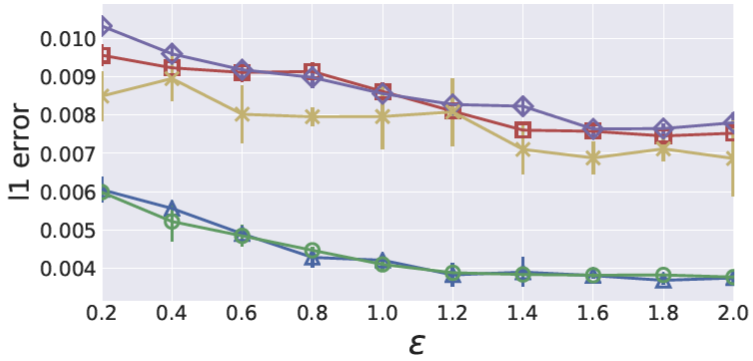


- **The performance of PrivBayes and PGM is close to PrivSyn for pair-wise marginal, meaning they can effectively capture low-dimensional correlation.**
- **PrivSyn significantly outperforms others for range query and classification, meaning PrivSyn can also preserve high-dimensional correlation.**

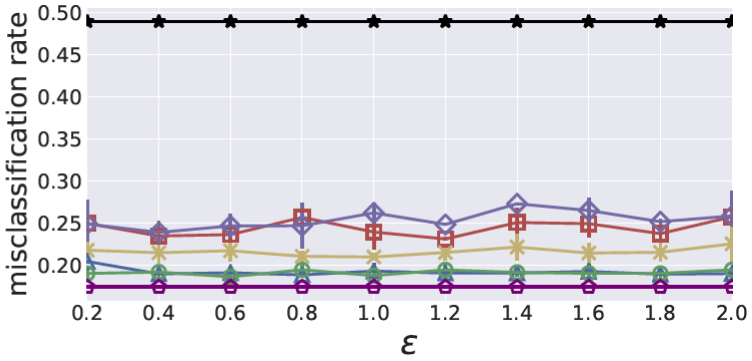
Marginal Selection Methods Comparison



Pair-wise marginal



Range query



Classification

DenseMarg (Non-private)
DenseMarg (Private)
PrivBayes(InDif) (Non-private)
PrivBayes(InDif) (Private)
Manual
Majority
NonPriv

- **DenseMarg consistently outperform PrivBayes (InDif) by exploiting more marginals.**
- **DenseMarg performs similar in the private setting and non-private (do not add noise in the marginal selection phase) setting, indicating DeseMarg is robust to noise.**

Open Questions

PGM (Esp. Markov Random Fields) vs Direct Data Synthesis

- Both methods extract information from input dataset via marginals
  - One uses them to build a PGM
  - The other uses them to directly synthesize data
- Jury is still out on the tradeoffs of using an PGM as an intermediate model
- Both methods can be further refined
  - How to select marginals?
  - How to synthesize datasets from marginals?
    - Extensive work in other community, e.g., transportation engineering

Better Ways to Deal with Numerical Attributes

- In current approaches, numerical attributes are binned into categorical values, which lose the ordered nature of them.

- There may be better ways to handle numerical attributes, exploiting the ordered nature

Paradoxical Gap Between Theory and Practice

- For complex tasks (such as DP Synthesis) any algorithm that has a theorem proving a bound of utility performs poorly in practice.
- That is, any algorithm that performs well in practice has no proof of utility bound.
- Corollary: Any proven utility bound is quite meaningless in practice.

Illustration of the Gap Using the MWEM paper

Inputs: Data set B over a universe D; Set Q of linear queries; Number of iterations $T \in \mathbb{N}$; Privacy parameter $\varepsilon > 0$; Number of records n.

Let A_0 denote n times the uniform distribution over D.
For iteration $i = 1, ..., T$:

1. *Exponential Mechanism:* Select a query $q_i \in Q$ using the Exponential Mechanism parameterized with epsilon value $\varepsilon/2T$ and the score function

$$s_i(B, q) = |q(A_{i-1}) - q(B)| \ .$$

2. *Laplace Mechanism:* Let measurement $m_i = q_i(B) + Lap(2T/\varepsilon)$.
3. *Multiplicative Weights:* Let A_i be n times the distribution whose entries satisfy

$$A_i(x) \propto A_{i-1}(x) \times \exp(q_i(x) \times (m_i - q_i(A_{i-1}))/2n) \ .$$

Output: $A = \text{avg}_{i<T} A_i$.

Figure 1: The MWEM algorithm.

Theorem 2.2. *For any dataset B, set of linear queries Q, $T \in \mathbb{N}$, and $\varepsilon > 0$, with probability at least $1 - 2T/|Q|$, MWEM produces A such that*

$$\max_{q \in Q} |q(A) - q(B)| \leq 2n\sqrt{\frac{\log |D|}{T}} + \frac{10T \log |Q|}{\varepsilon} \ .$$

Inputs: Data set B over a universe D; Set Q of linear queries; Number of iterations $T \in \mathbb{N}$; Privacy parameter $\varepsilon > 0$; Number of records n.

Let A_0 denote n times the uniform distribution over D.
For iteration $i = 1, ..., T$:

1. *Exponential Mechanism:* Select a query $q_i \in Q$ using the Exponential Mechanism parameterized with epsilon value $\varepsilon/2T$ and the score function

$$s_i(B, q) = |q(A_{i-1}) - q(B)| \ .$$

2. *Laplace Mechanism:* Let measurement $m_i = q_i(B) + Lap(2T/\varepsilon)$.
3. *Multiplicative Weights:* Let A_i be n times the distribution whose entries satisfy

$$A_i(x) \propto A_{i-1}(x) \times \exp(q_i(x) \times (m_i - q_i(A_{i-1}))/2n) \ .$$

Output: $A = \text{avg}_{i<T} A_i$.

Figure 1: The MWEM algorithm.

2.3.2 Improvements and Variations

There are several ways to improve the empirical performance of MWEM at the expense of the theoretical guarantees. First, rather than use the average of the distributions A_i we use only the final distribution. Second, in each iteration we apply the multiplicative weights update rule for all measuments taken, multiple times; as long as any measurements do not agree with the approximating distribution (within error) we can improve the result. Finally, it is occasionally helpful to initialize A_0 by performing a noisy count for each element of the domain; this consumes from the privacy budget and lessens the accuracy of subsequent queries, but is often a good trade-off.

Possible Reasons Behind Paradoxical Gap

- Practically effective algorithms are complex and thus difficult to analyze

- Proven utility bounds must hold for the worst-case datasets (including pathological ones)
  - Can there be theory of average-case behavior, studying effects of an algorithm on “common” datasets (or distributions)

Thank you for your listening

Q & A

Email: ninghui@purdue.edu

Website: https://www.cs.purdue.edu/homes/ninghui/