PrivSyn: Differentially Private Data Synthesis

Presenter: Zhikun Zhang

Joint work with Tianhao Wang, Ninghui Li, Jean Honorio,

Michael Backes, Shibo He, Jiming Chen, and Yang Zhang

浙江大学 Zhejiang University

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

PURDUE UNIVERSITY

Outline

- ❑ Background and Preliminaries
- ❑ Existing Work
- ❑ Our Proposal
- ❑ Experimental Evaluation
- ❑ Conclusion

Outline

- ❑ Background and Preliminaries
- ❑ Existing Work
- ❑ Our Proposal
- ❑ Experimental Evaluation
- ❑ Conclusion

Big Data Era

- Data collection
  - Browsing history, typing habit
- Data analysis
  - Improve user exprience, recomemdation

User Data

Data Collection

Data Analysis

Privacy Accidents

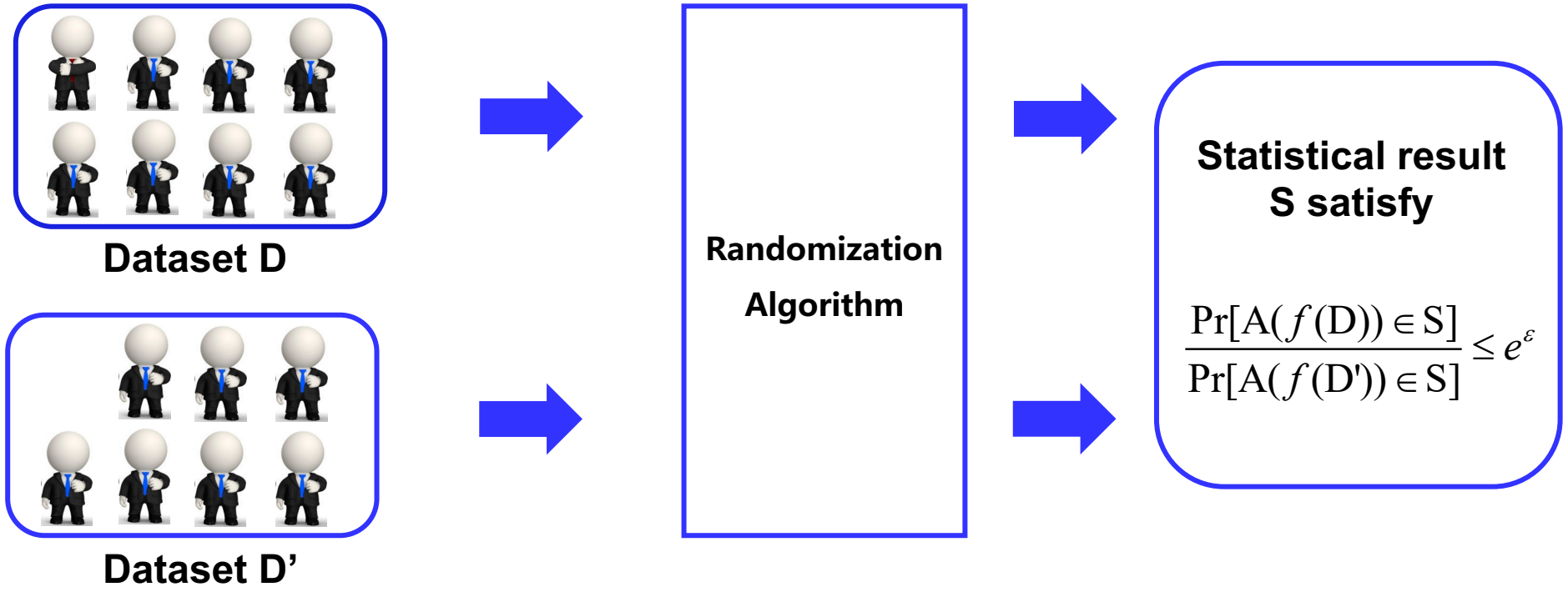
2018, Huazhu Hotel
breach 0.5 billion user data

2017, Yahoo
breach 3 billion user data

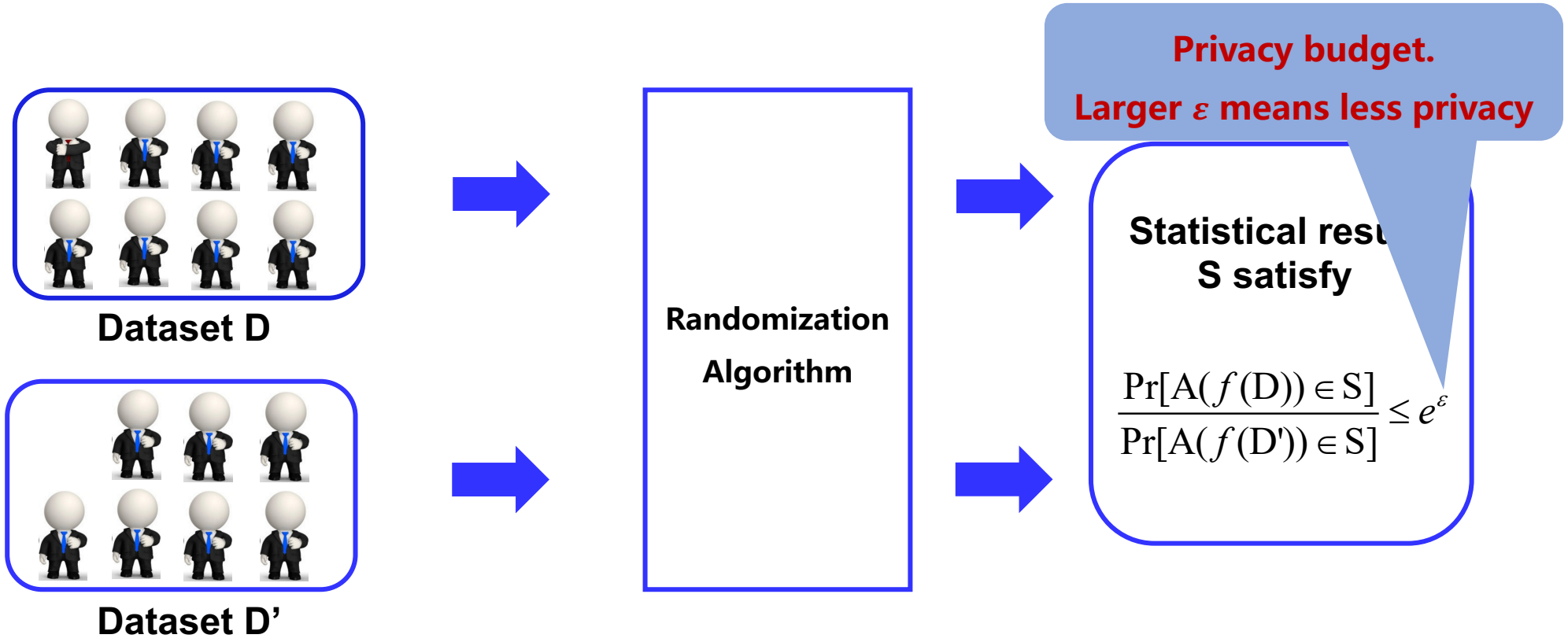
2016, MySpace
breach 3.6 billion user accounts

2013, Snowden
exposed the PRISM program

Differential Privacy (DP)



Differential Privacy (DP)



Laplace Mechanism

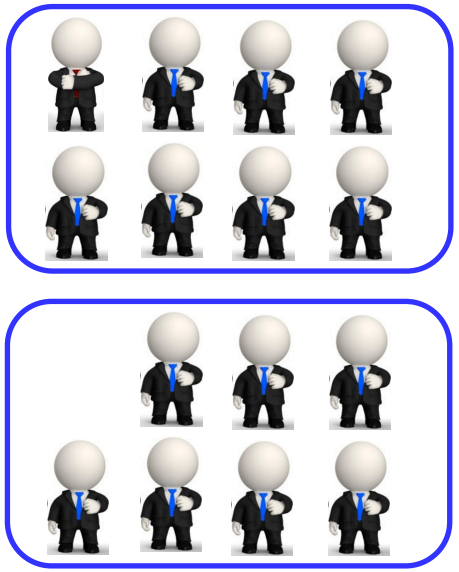
Randomized value

Sensitivity

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

Real value

$$\Delta_f = \max ||f(D) - f(D')||$$



8

7

$$\Delta_f = 8 - 7 = 1$$

Gaussian Mechanism

$$\mathrm{A}(D) = f(D) + N(0, (\frac{\Delta_f}{\varepsilon})^2)$$

Gaussian mechanism only satisfies (ε, δ)-DP

- **Violate ϵ-DP with very small probability δ**
- **δ is always set to be $\frac{1}{n^2}$**

Approximate Differential Privacy

Outline

- ❑ Background and Preliminaries
- ❑ Existing Work
- ❑ Our Proposal
- ❑ Experimental Evaluation
- ❑ Conclusion

Tailored Algorithms

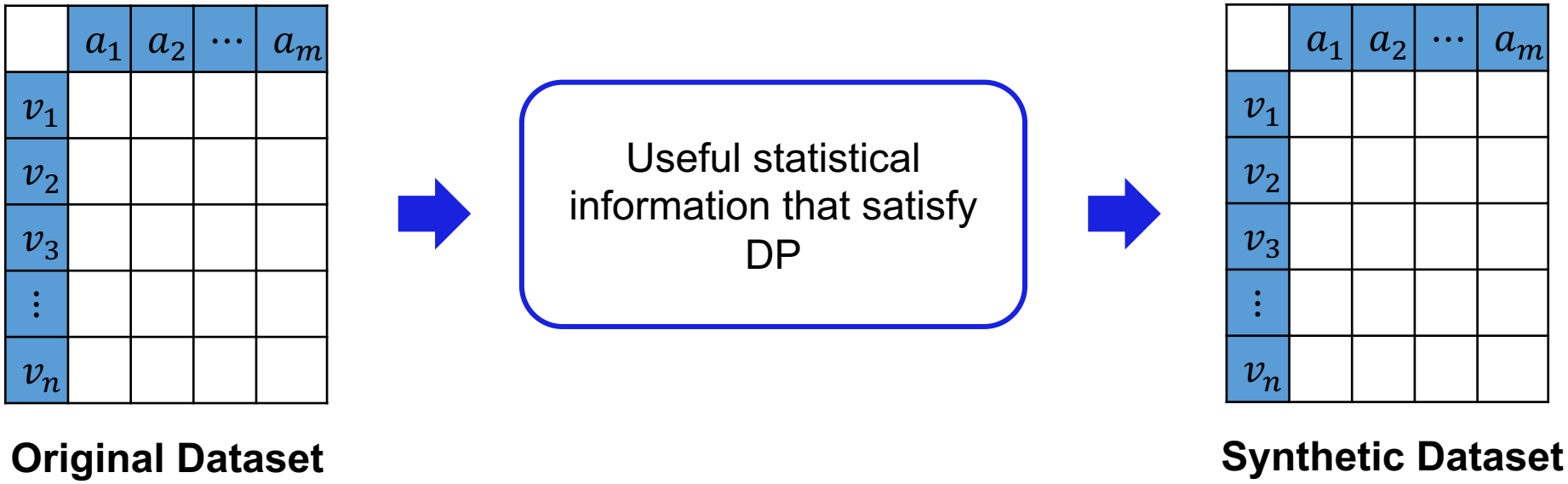
- Frequent itemset mining [VLDB'12], [KDD'14], [ICDE'17]
- Marginal release [PODS'07], [SIGMOD'14]
- Range query [TKDE'10], [ICDE'12]
- Machine learning models [CCS'16], [S&P'19]

Tailored Algorithms

- Frequent itemset mining [VLDB'12], [KDD'14], [ICDE'17]
- Marginal release [PODS'07], [SIGMOD'14]
- Range query [TKDE'10], [ICDE'12]
- Machine learning models [CCS'16], [S&P'19]

- **Time consuming**
- **Requires a lot of expertise knowledge**
- **Error-prone**

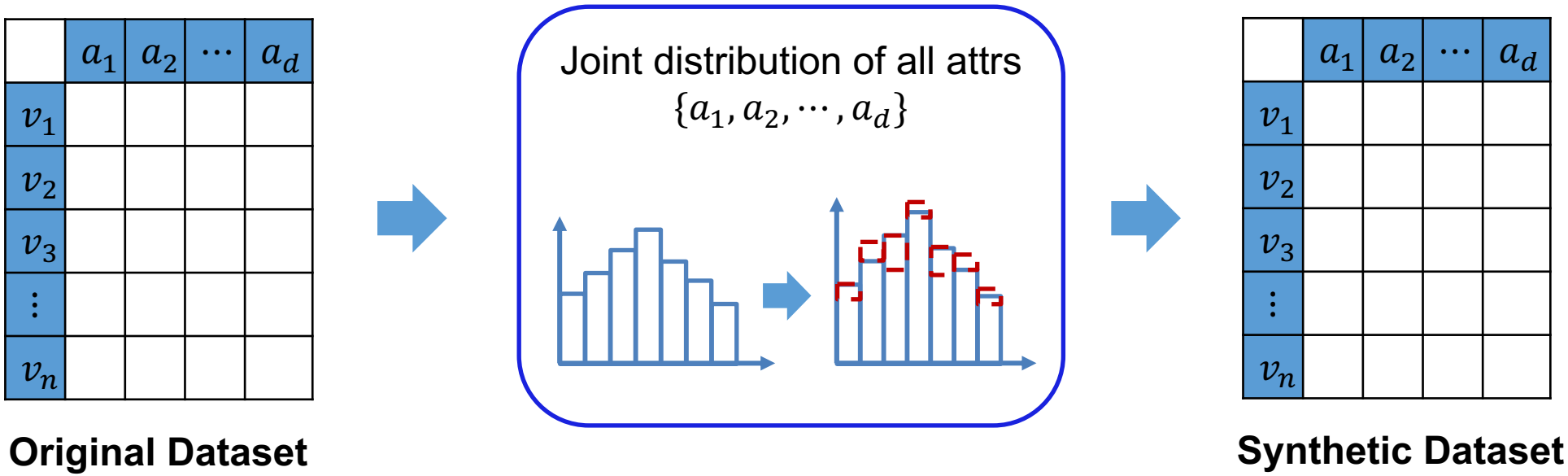
General Solution: Data Synthesis



Original Dataset

Synthetic Dataset

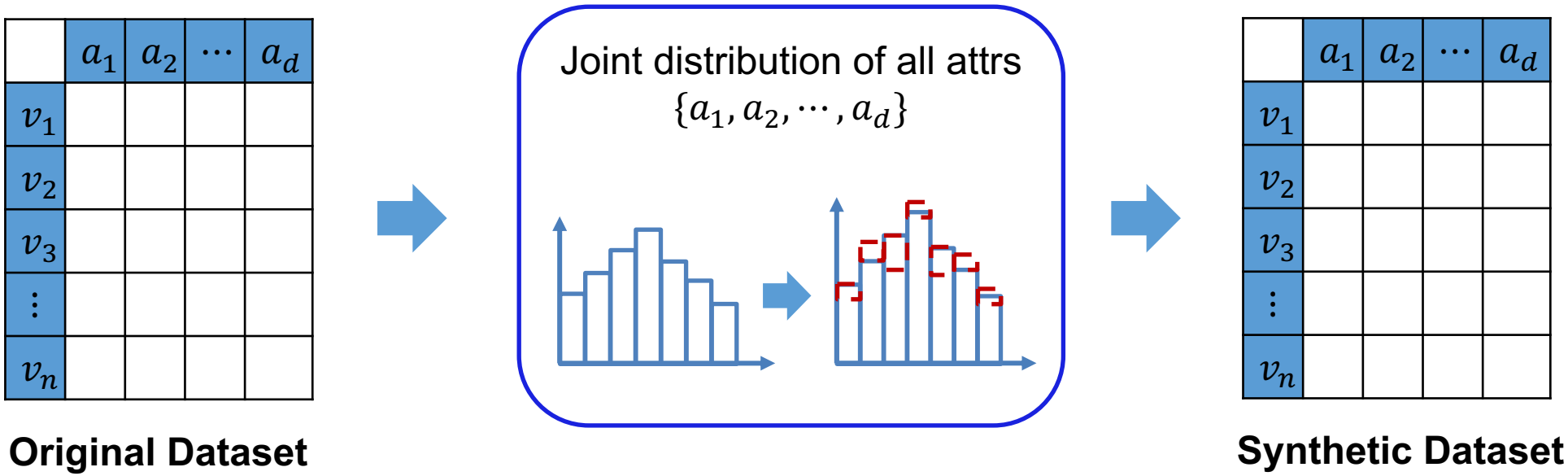
Naïve Method



Original Dataset

Synthetic Dataset

Naïve Method

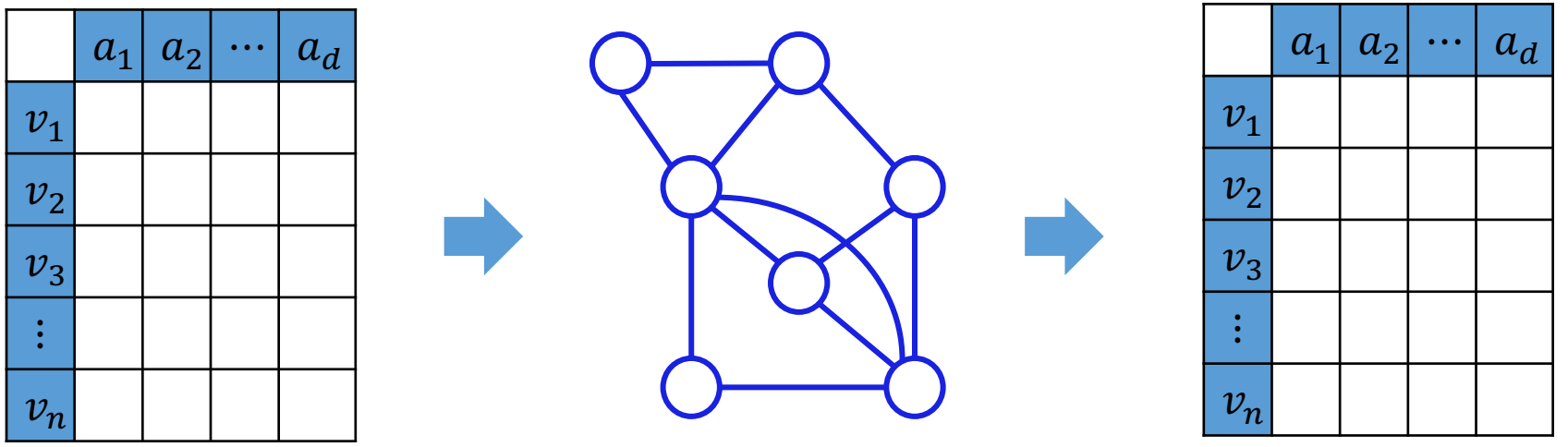


Original Dataset

Synthetic Dataset

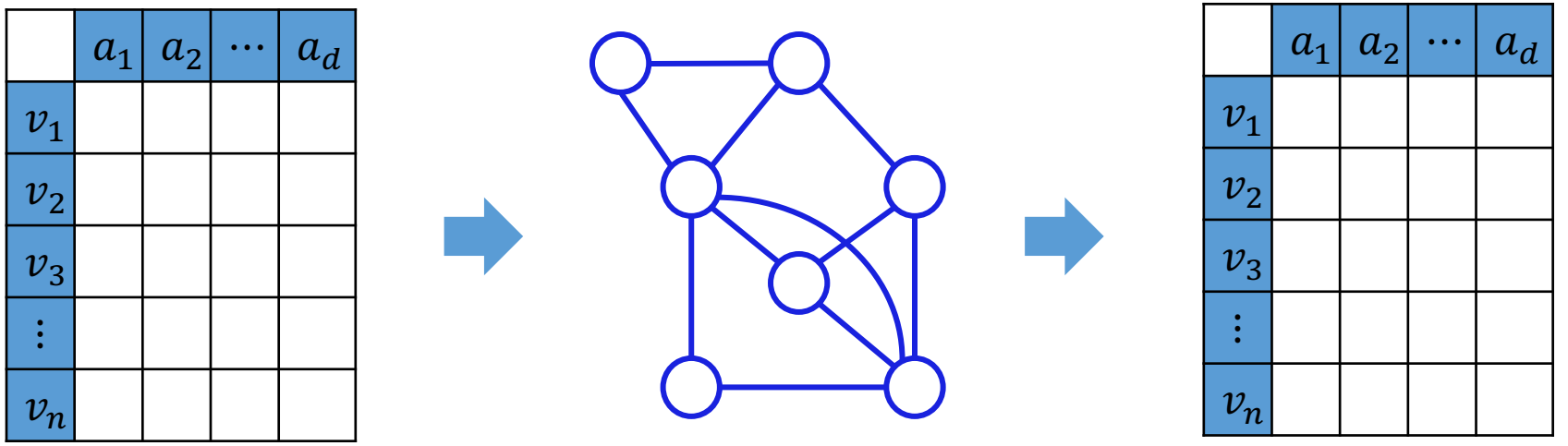
❖ **When the number of attributes is large, the domain of joint distribution is large, leading to prohibitive computational cost.**

Existing Method: Graphical Model



[1] 2014 SIGMOD PrivBayes: Private data release via Bayesian networks
[2] 2019 ICML Graphical model based estimation and inference for differential privacy

Existing Method: Graphical Model



- **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.**

[1] 2014 SIGMOD PrivBayes: Private data release via Bayesian networks
[2] 2019 ICML Graphical model based estimation and inference for differential privacy

Existing Method: Generative Model

Original Dataset **Synthetic Dataset**

[1] https://www.challenge.gov/challenge/differential-privacy-synthetic-data-challenge/

Existing Method: Generative Model

Original Dataset **Synthetic Dataset**

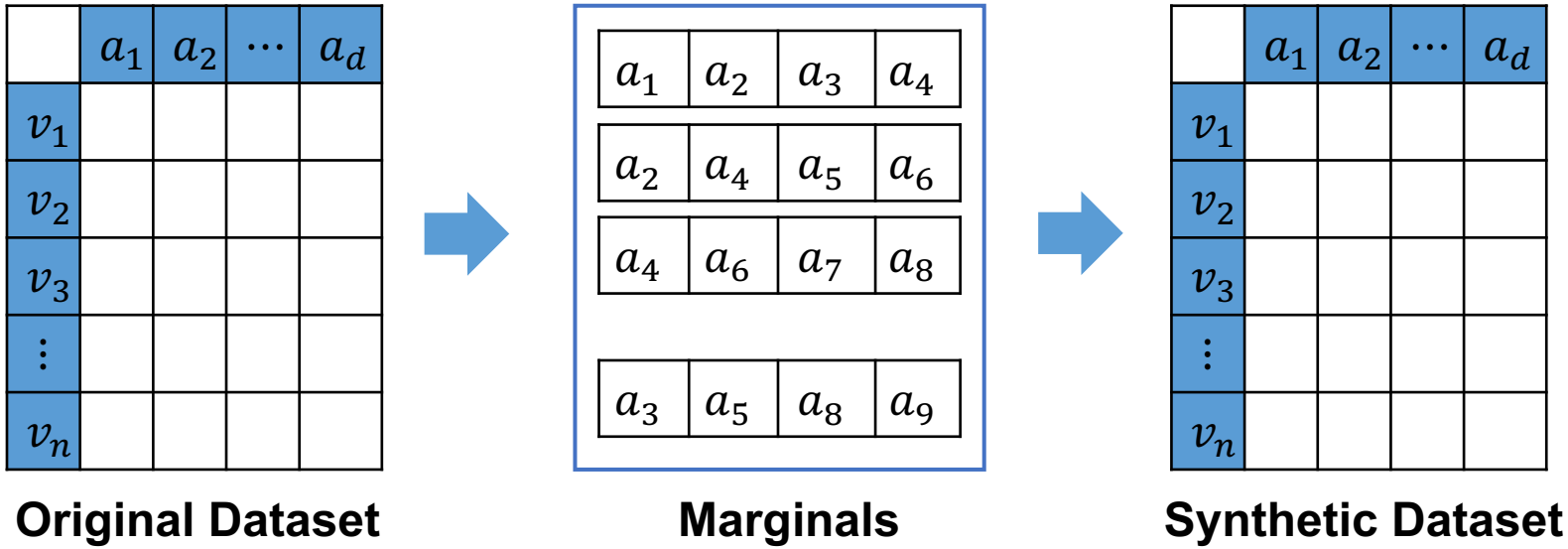
- **Performs well on image dataset.**
- **Cannot achieve satisfiable performance for high-dimensional tabular data. [1]**

[1] https://www.challenge.gov/challenge/differential-privacy-synthetic-data-challenge/

Outline

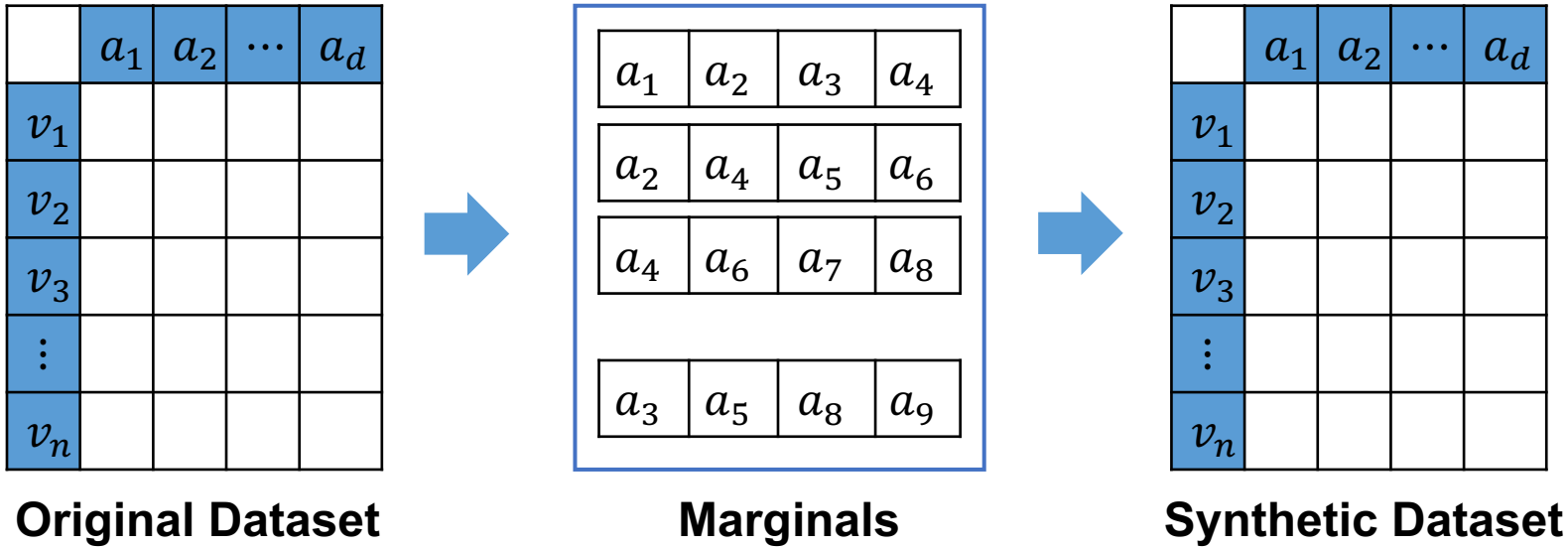
- ❑ Background and Preliminaries
- ❑ Existing Work
- ❑ Our Proposal
- ❑ Experimental Evaluation
- ❑ Conclusion

PrivSyn Overview



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

PrivSyn Overview



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

Challenge 1: How to choose a set of marginals that captures 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)

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)

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

	a_1	a_2	a_3	a_4	$\cdots$	a_d
v_1						
v_2						
v_3						
$\vdots$						
v_n						

Step I: Randomly generate dataset.

Dataset Generation: GUM

	a_1	a_2	a_3	a_4	$\cdots$	a_d
v_1						
v_2						
v_3						
$\vdots$						
v_n						

	a_1	a_2	a_3	a_4	$\cdots$	a_d
v_1						
v_2						
v_3						
$\vdots$						
v_n						

Step I: Randomly generate dataset.

Step II: Using all marginals to update the randomly generated dataset.

Dataset Generation: GUM

	a_1	a_2	a_3	a_4	$\cdots$	a_d
v_1						
v_2						
v_3						
$\vdots$						
v_n						

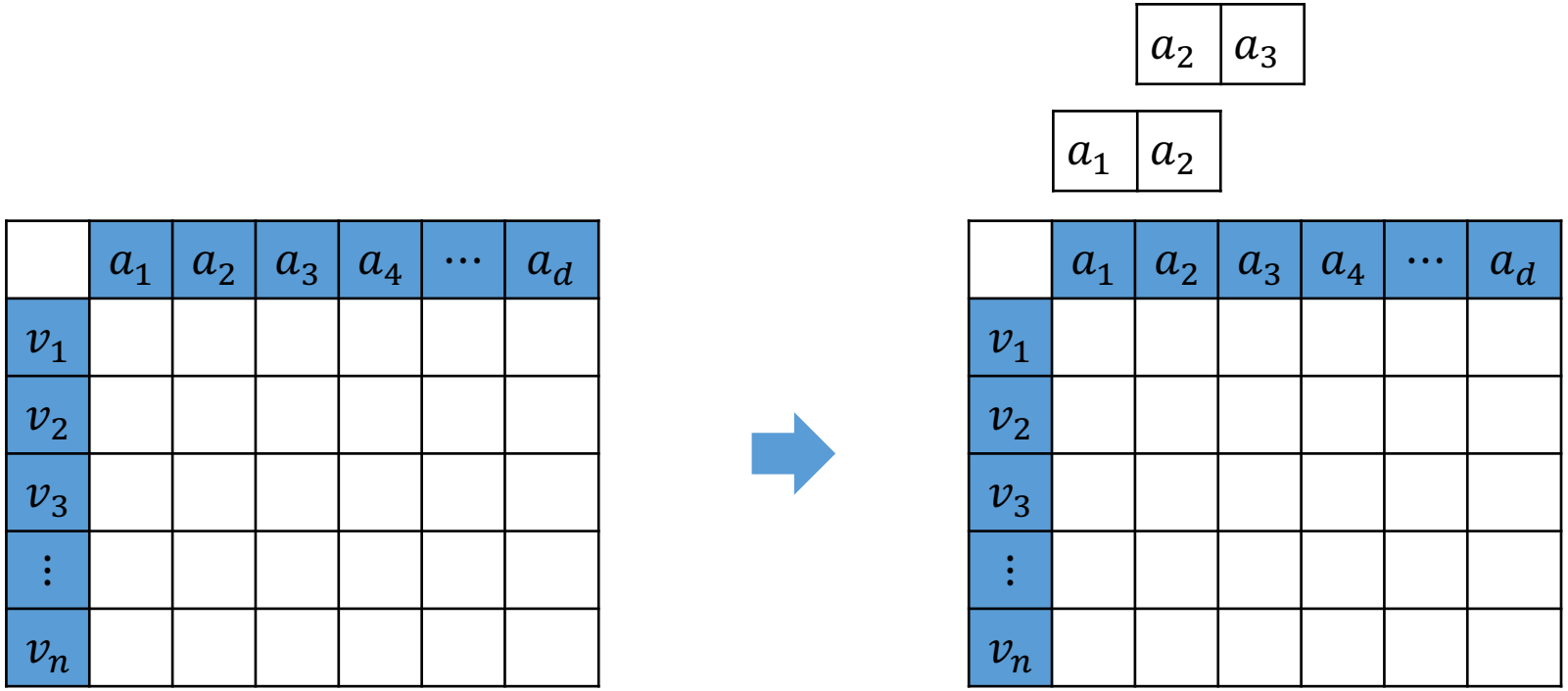
a_1	a_2

	a_1	a_2	a_3	a_4	$\cdots$	a_d
v_1						
v_2						
v_3						
$\vdots$						
v_n						

Step I: Randomly generate dataset.

Step II: Using all marginals to update the randomly generated dataset.

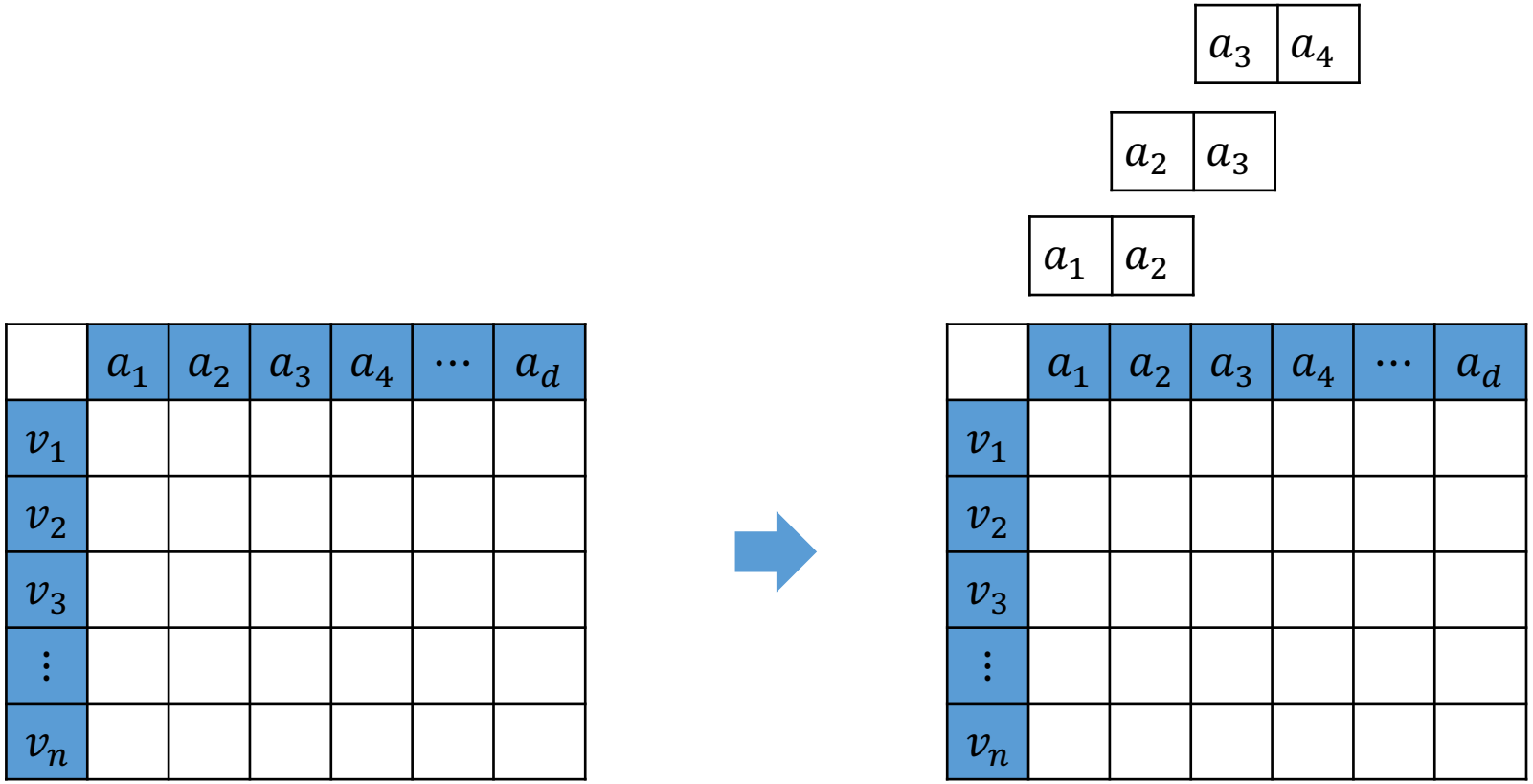
Dataset Generation: GUM



Step I: Randomly generate dataset.

Step II: Using all marginals to update the randomly generated dataset.

Dataset Generation: GUM



Step I: Randomly generate dataset.

Step II: Using all marginals to update the randomly generated dataset.

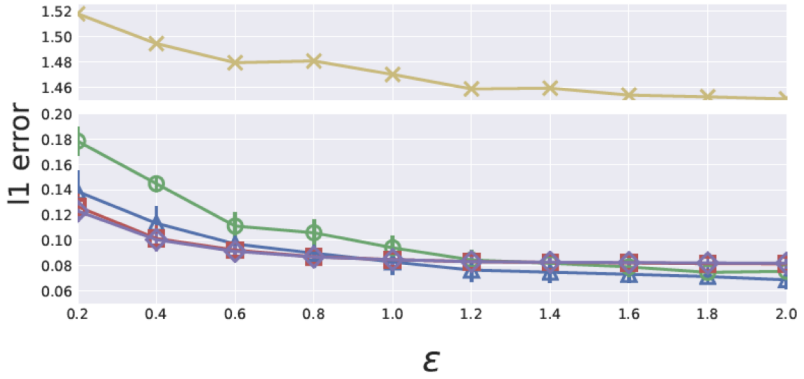
Outline

- ❑ Background and Preliminaries
- ❑ Existing Work
- ❑ Our Proposal
- ❑ Experimental Evaluation
- ❑ Conclusion

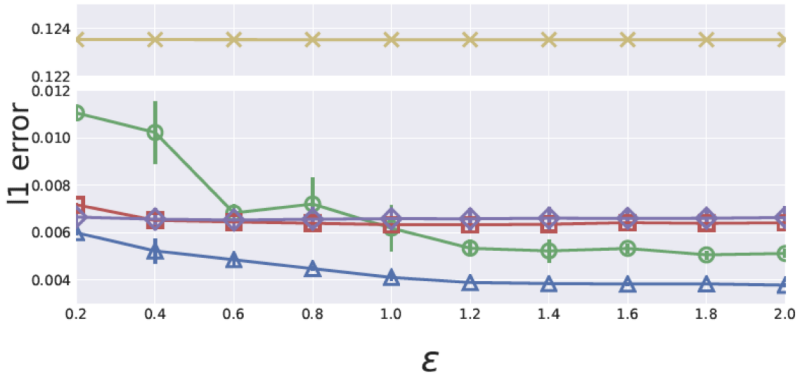
Experimental Setup

- 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)

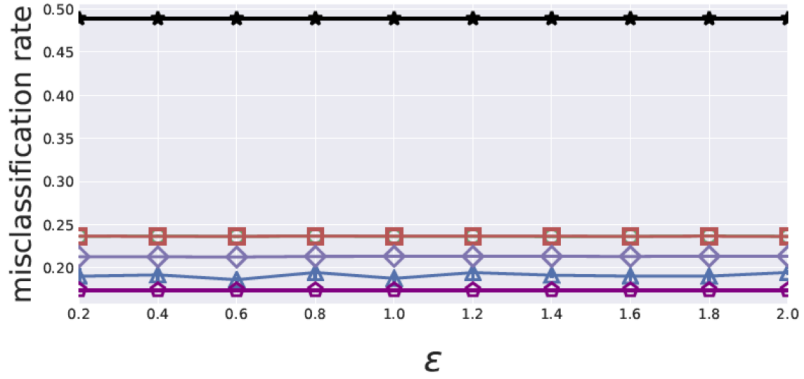
End-to-end Comparison



Pair-wise marginal



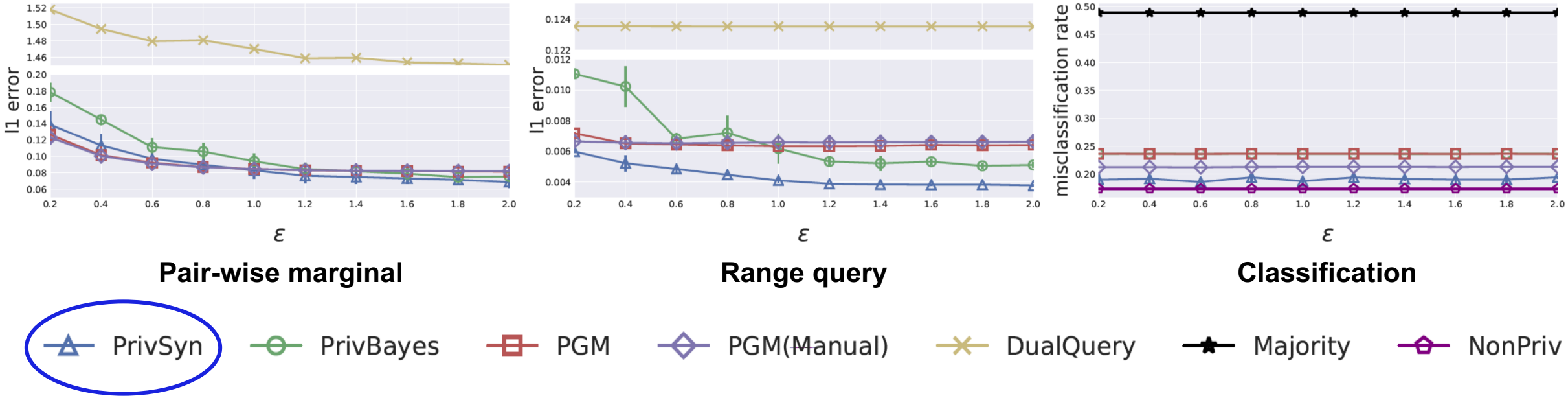
Range query



Classification

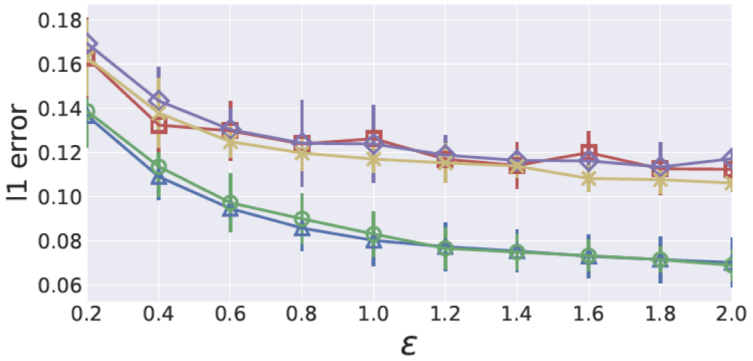
PrivSyn PrivBayes PGM PGM(Manual) DualQuery Majority NonPriv

End-to-end Comparison

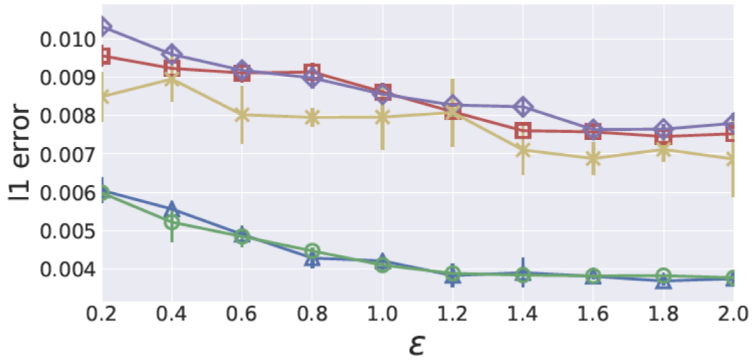


- **The performance of PrivBayes and PGM is close to PrivSyn for pair-wise marginal, meaning they can effectively capture low-dimensional correlation.**
- **PrivSyn significantly outperforms others for range query and classification, meaning PrivSyn can also preserve high-dimensional correlation.**

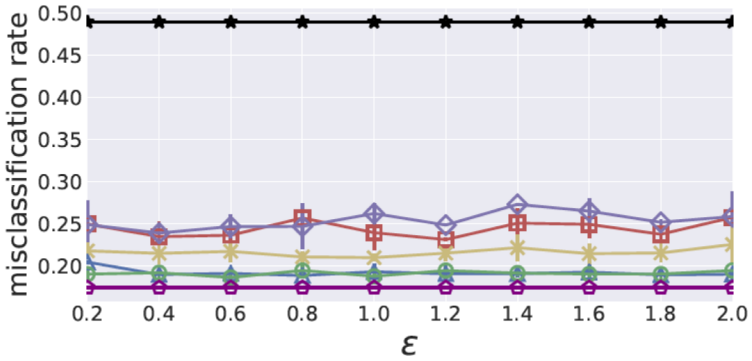
Marginal Selection Methods Comparison



Pair-wise marginal



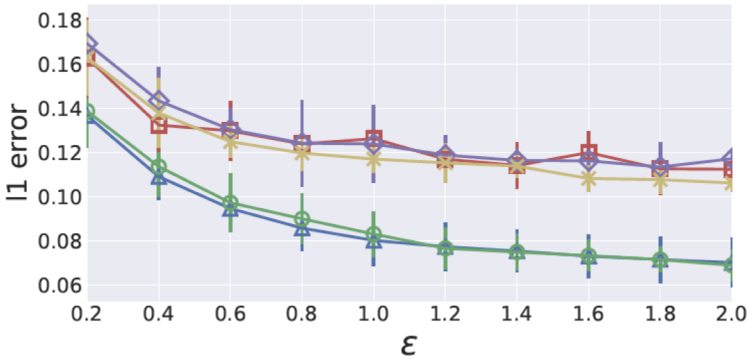
Range query



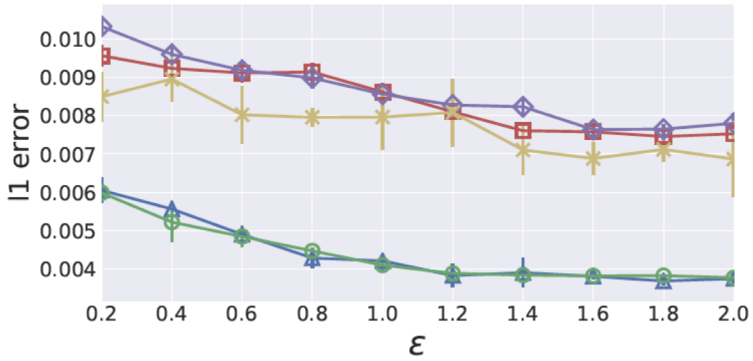
Classification

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

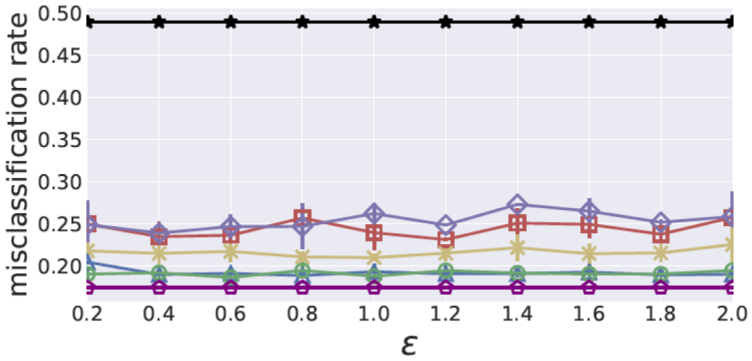
Marginal Selection Methods Comparison



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.**

Outline

- ❑ Background and Preliminaries
- ❑ Existing Work
- ❑ Our Proposal
- ❑ Experimental Evaluation
- ❑ Conclusion

Conclusion

- A new method to automatically and privately select marginals that capture sufficient correlations.
- A data synthesis algorithm that can also be used standalone to handle dense graphical models.
- An extensive evaluation which demonstrates the performance improvement of the proposed method on real-world datasets and helps us understand the intuition of different techniques.

Thank you for your listening

Q & A

Email: zhikun.zhang@cispa.de

Website: www.zhangzhk.com

Twitter: @ZhikunZhang5