Co-clustering for differentially private synthetic data generation

Tarek Benkhelif, Françoise Fessant, Fabrice Clérot and Guillaume Raschia

January 23, 2018

Orange Labs & LS2N

Journée thématique EGC & IA : Données personnelles, vie privée et éthique

Context

Privacy preserving data publishing

- Releasing data, either in their `original` or `aggregated` form
- Protecting individuals represented in the data
- Providing sufficient utility

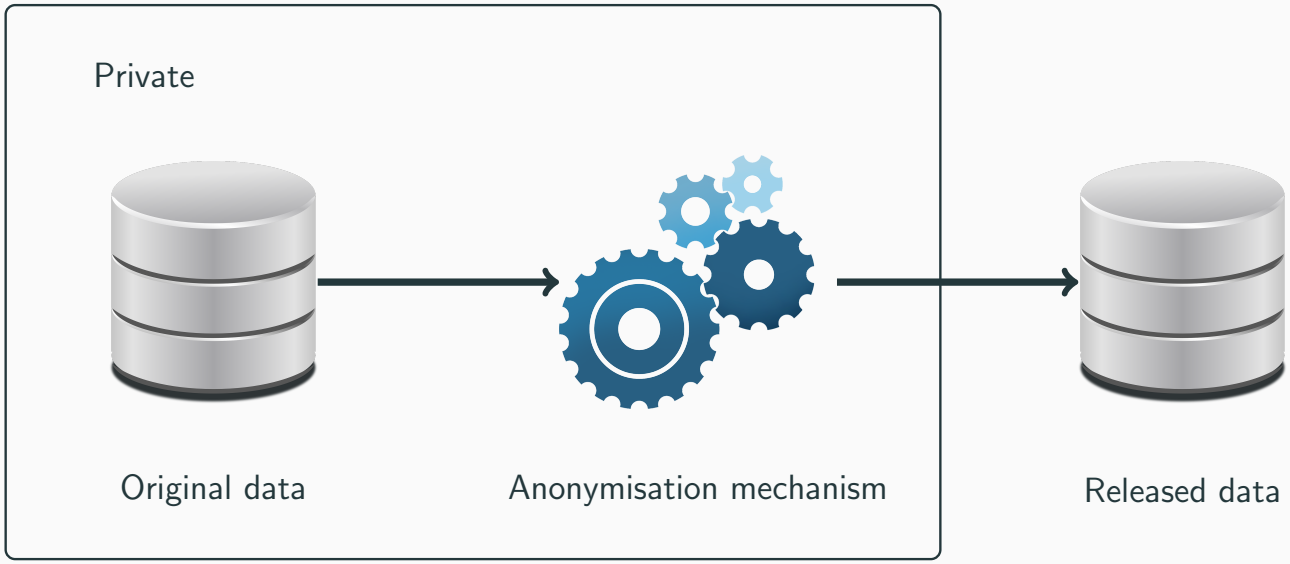
Privacy preserving data publishing

Original data

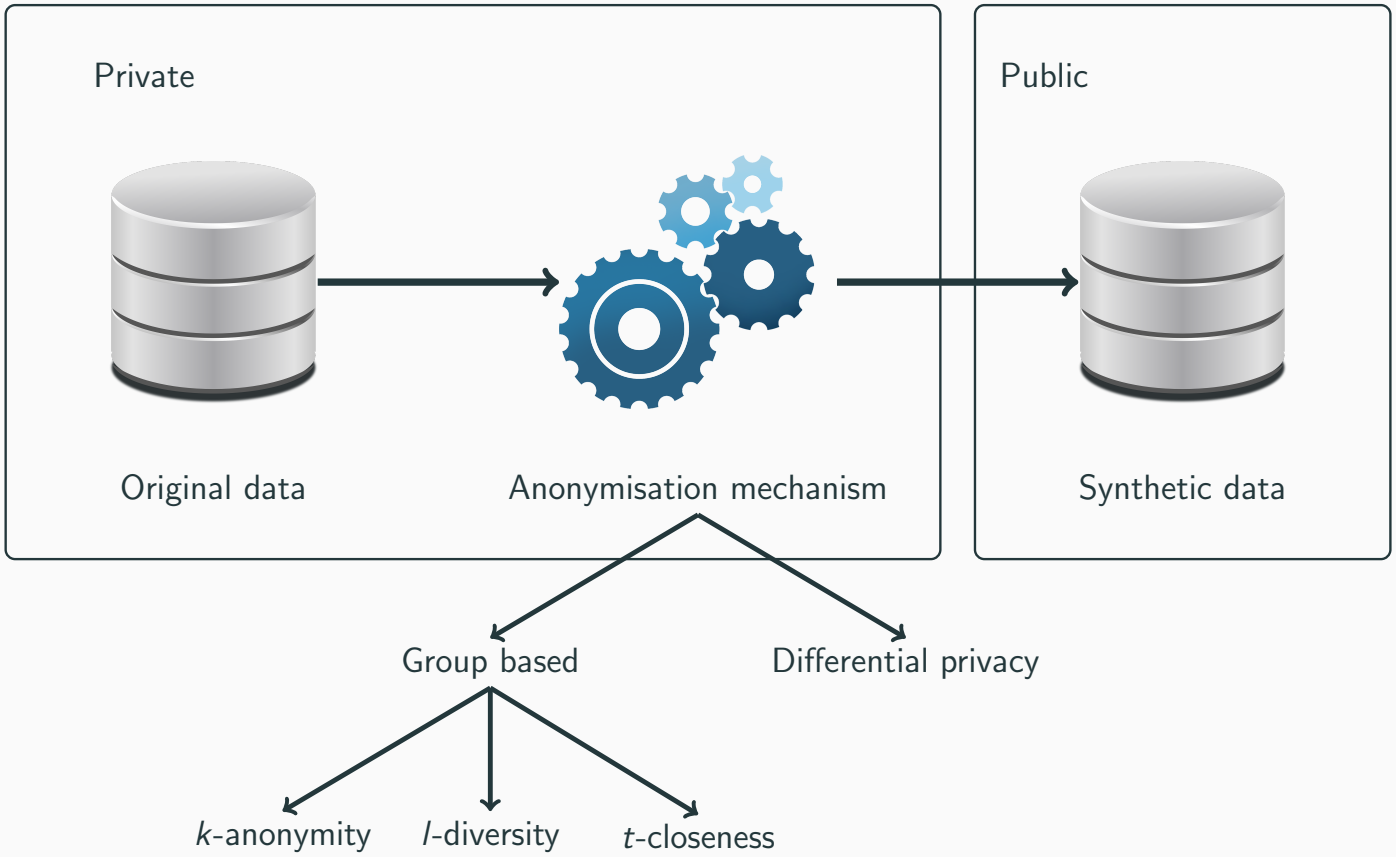
Anonymisation mechanism

Released data

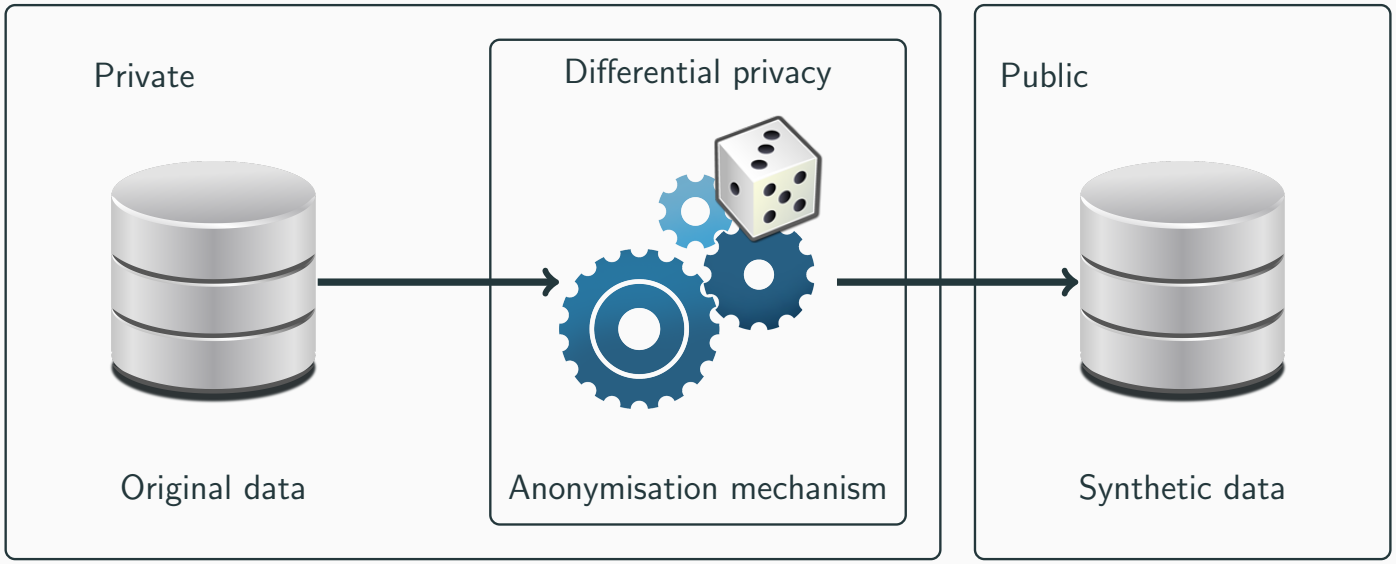
Privacy preserving data publishing



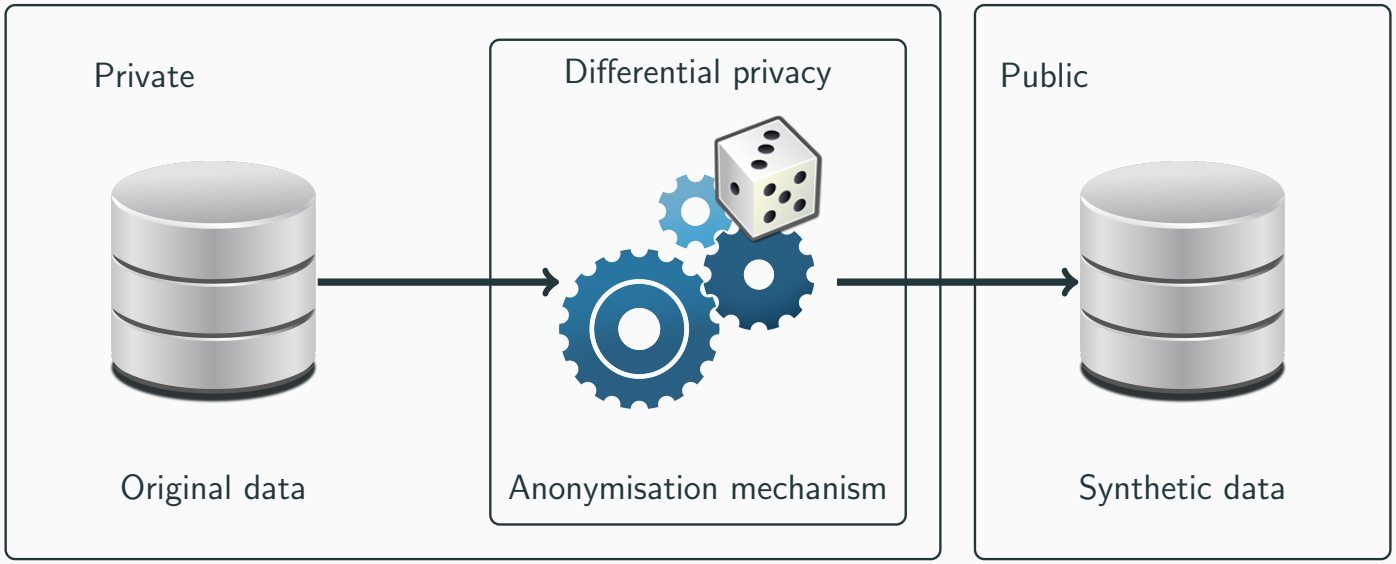
Privacy preserving data publishing



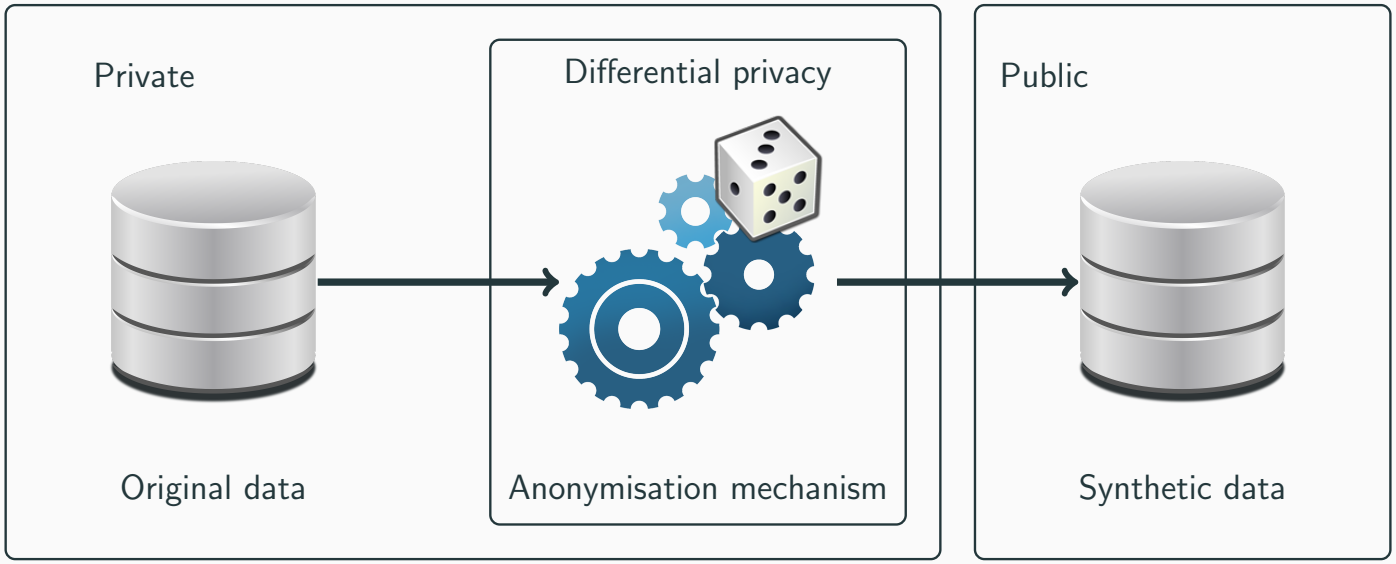
Privacy preserving data publishing



Privacy preserving data publishing

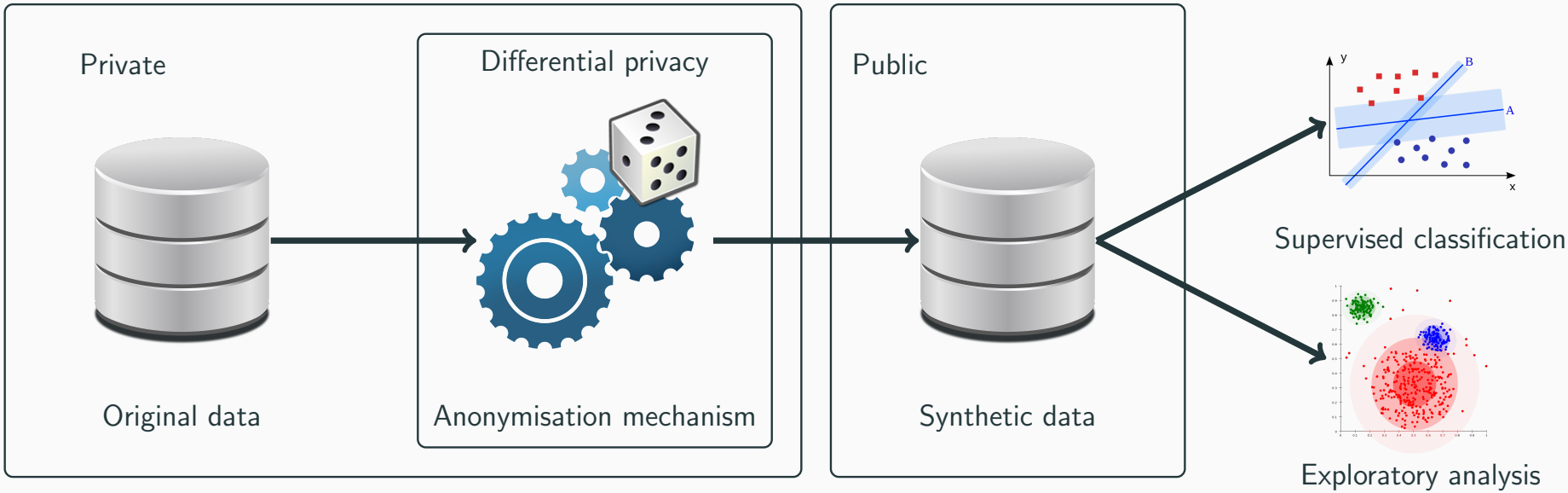


Privacy preserving data publishing



- Same format as the original data
- Multidimensional data
- Independent of the data mining task

Privacy preserving data publishing



- Same format as the original data
- Multidimensional data
- Independent of the data mining task

Differential Privacy: Intuition

Differential Privacy

- It should not harm you or help you as an individual to enter or to leave the dataset.
- To ensure this property, we need a mechanism whose output is nearly unchanged by the presence or absence of a single respondent in the database.
- In constructing a formal approach, we concentrate on pairs of databases (D_1, D_2) differing on only one row, with one a subset of the other and the larger database containing a single additional row.

Differential Privacy

ε-Differential Privacy [Dwo06]

A data release mechanism $\mathcal{A}$ satisfies ε-differential privacy if for all neighboring database D_1 and D_2, and released output O,

$Pr[\mathcal{A}(D_1) = O] \leq e^{\varepsilon} \times Pr[\mathcal{A}(D_2) = O]$.

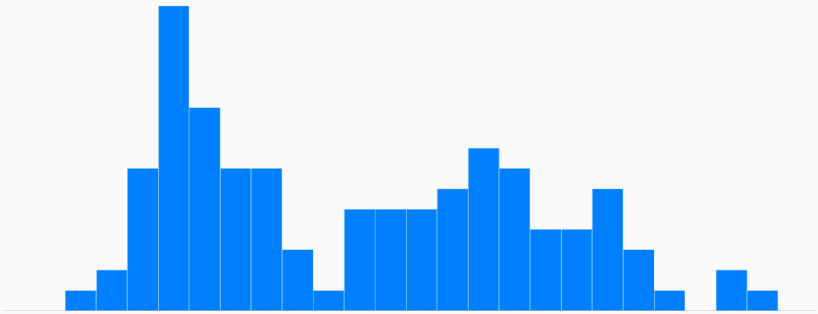
Achieving ε-DP : Laplace mechanism

Adds random noise to the true answer of a query Q,

$\mathcal{A}_Q(D) = Q(D) + \tilde{N}$, where $\tilde{N}$ is the Laplace noise. The magnitude of the noise depends on the privacy levels and the query's sensitivity

Existing approaches

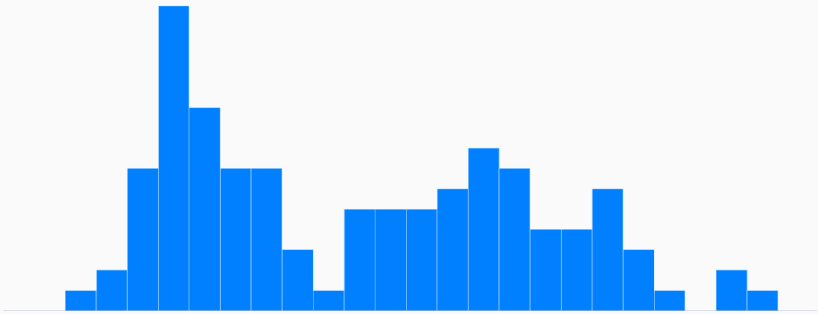
Base line algorithm



1. Discretize attribute domain into cells

Limitations

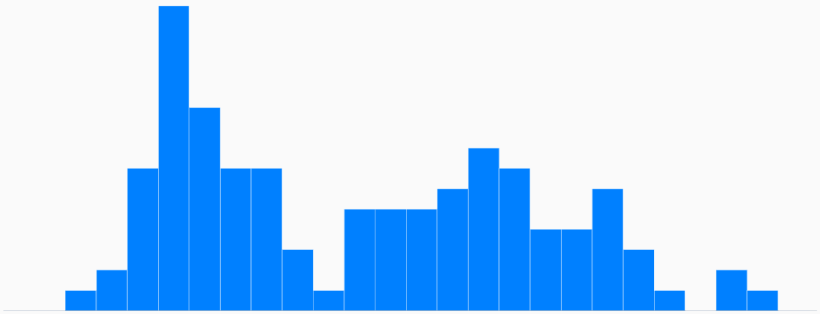
Base line algorithm



1. Discretize attribute domain into cells
2. Add noise to cell counts (Laplace mechanism)

Limitations

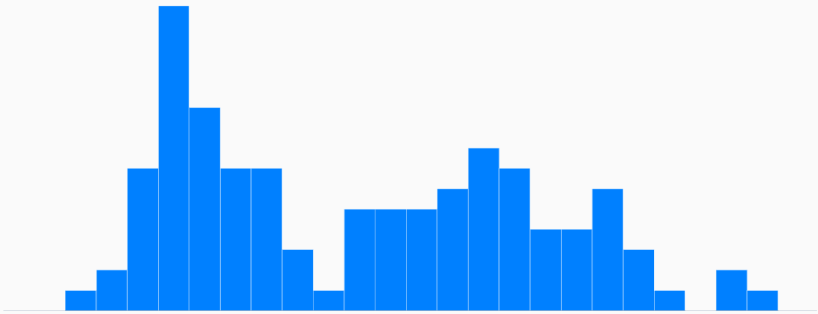
Base line algorithm



1. Discretize attribute domain into cells
2. Add noise to cell counts (Laplace mechanism)
3. Use noisy counts to either...

Limitations

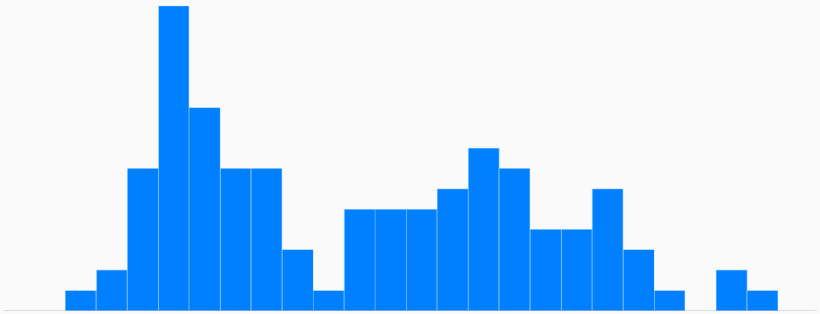
Base line algorithm



Limitations

1. Discretize attribute domain into cells
2. Add noise to cell counts (Laplace mechanism)
3. Use noisy counts to either...
   - 3.1 Answer queries directly (assume distribution is uniform within cell)

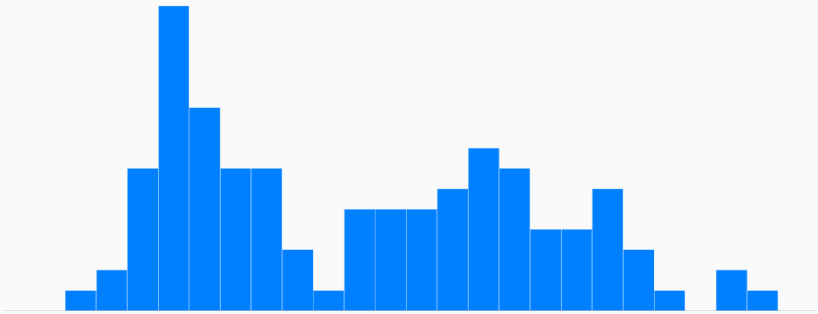
Base line algorithm



Limitations

1. Discretize attribute domain into cells
2. Add noise to cell counts (Laplace mechanism)
3. Use noisy counts to either...
   - 3.1 Answer queries directly (assume distribution is uniform within cell)
   - 3.2 Generate synthetic data (derive distribution from counts and sample)

Base line algorithm

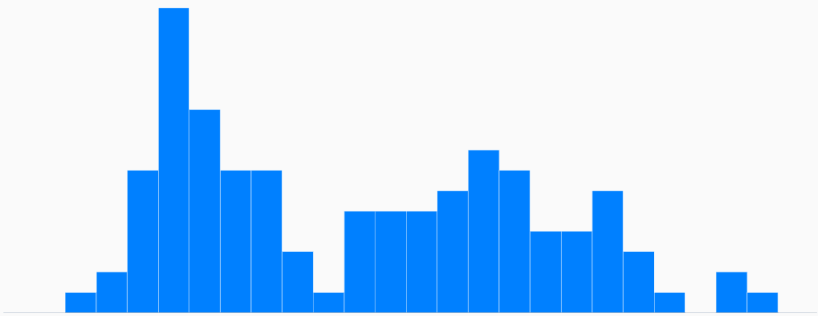


Limitations

- Granularity of discretization

1. Discretize attribute domain into cells
2. Add noise to cell counts (Laplace mechanism)
3. Use noisy counts to either...
   - 3.1 Answer queries directly (assume distribution is uniform within cell)
   - 3.2 Generate synthetic data (derive distribution from counts and sample)

Base line algorithm

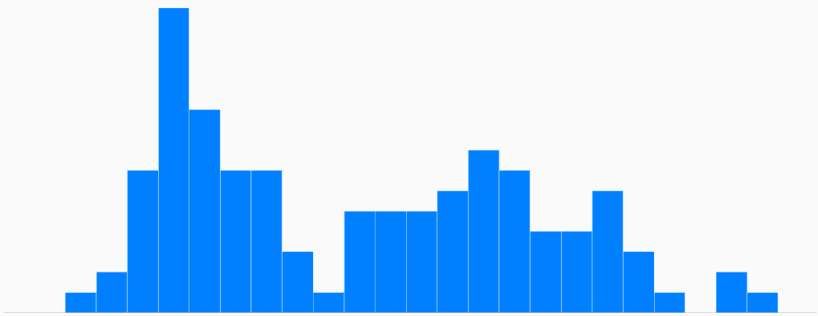


Limitations

- Granularity of discretization
  - Coarse: detail lost

1. Discretize attribute domain into cells
2. Add noise to cell counts (Laplace mechanism)
3. Use noisy counts to either...
   - 3.1 Answer queries directly (assume distribution is uniform within cell)
   - 3.2 Generate synthetic data (derive distribution from counts and sample)

Base line algorithm



Limitations

- Granularity of discretization
  - Coarse: detail lost
  - Fine: noise overwhelms signal

1. Discretize attribute domain into cells
2. Add noise to cell counts (Laplace mechanism)
3. Use noisy counts to either...
   - 3.1 Answer queries directly (assume distribution is uniform within cell)
   - 3.2 Generate synthetic data (derive distribution from counts and sample)

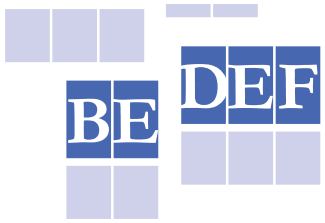
DP multidimensional data release approaches

Approach	Dimension	Mixed data type	Parameter(s)
DPCube [XXFG12]	Multi-D	✗	Variance threshold
DP-MHMD [RKS16]	Multi-D	✗	Attribute grouping
DiffGen [MCFY11]	Multi-D	✓	• Attributes taxonomy • n^{br} of specializations
PrivBayes [ZCP$^+$14]	Multi-D	✓	Bayesian network degree

PrivBayes [ZCP+14]

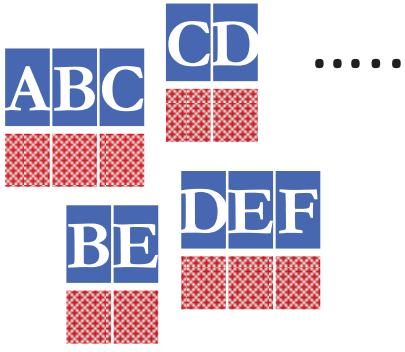
High-dimensional table T

decompose



Low-dimensional tables

Add noise



Noisy tables

- Method:
  - Use **Bayesian network** to learn data distribution
  - After BN learned, generate *synthetic data* by sampling from BN
- **Challenge**: *privately* choosing *good* decomposition

Noisy table T^*

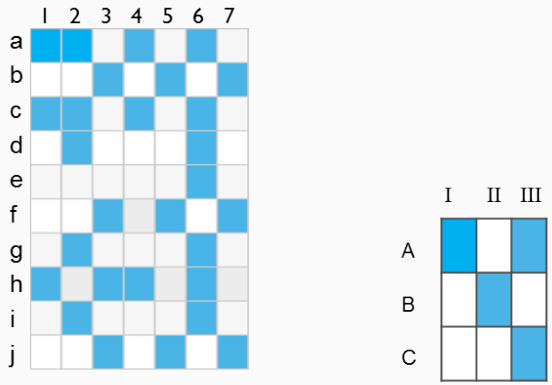
reconstruct

Proposition: DPCocGen

Co-clustering

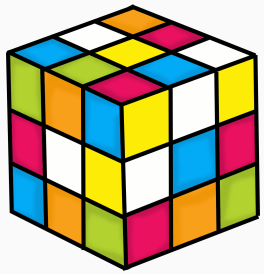
Bi-clustering

Simultaneously partition the rows and columns of a data matrix.



D-clustering

Simultaneously partition the d-dimensions of a data hyper cube.



Capture the interaction (underlying structure) between the d entities.

MODL Co-clustering features

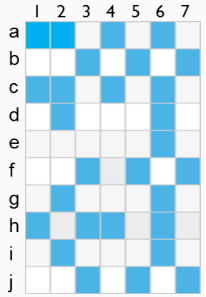
Grouping

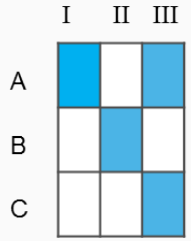
Discover the best reordering and grouping of the data cube [1] that:

- maximize the mutual information between the d-clusterings

Aggregation

Aggregation ability which allows to decrease the number of clusters in a greedy optimal way



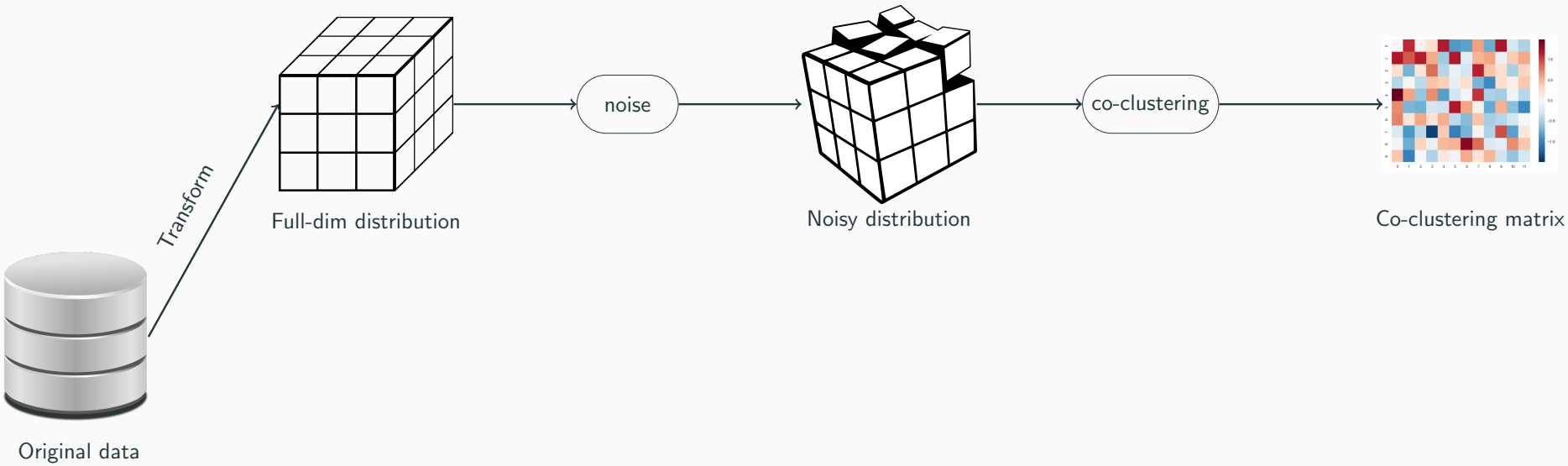


[1] Boullé , M.: Functional data clustering via piecewise constant nonparametric density estimation.

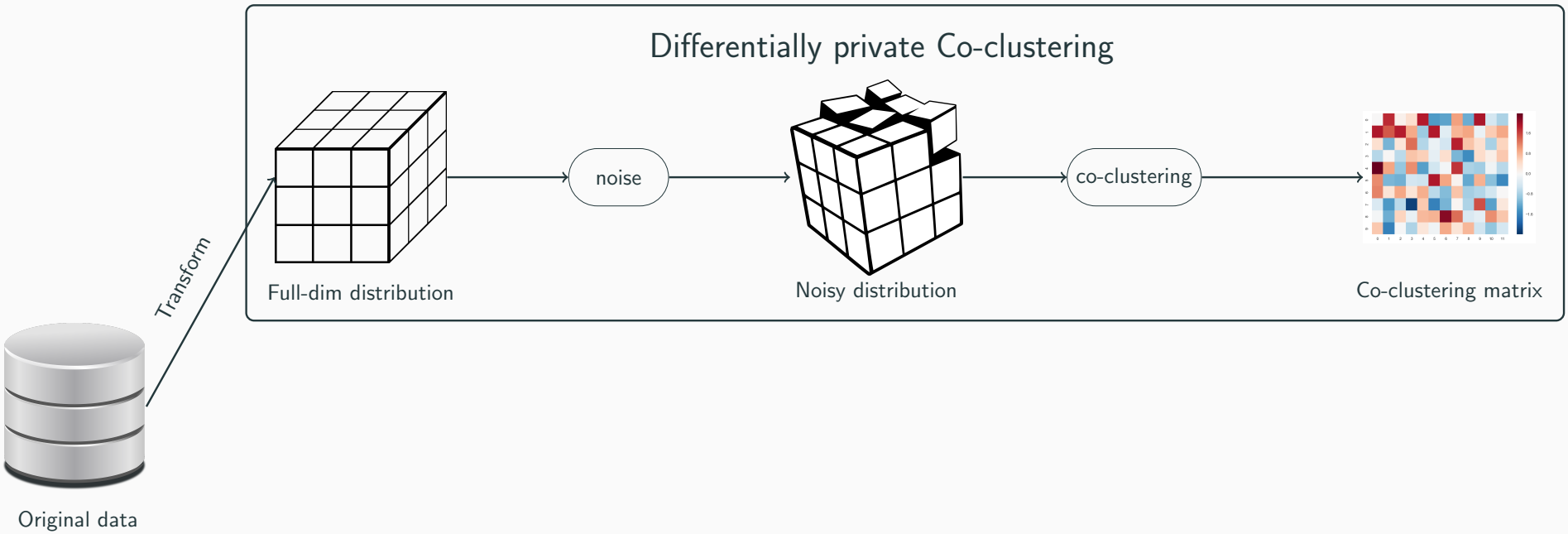
DPCocGen

Original data

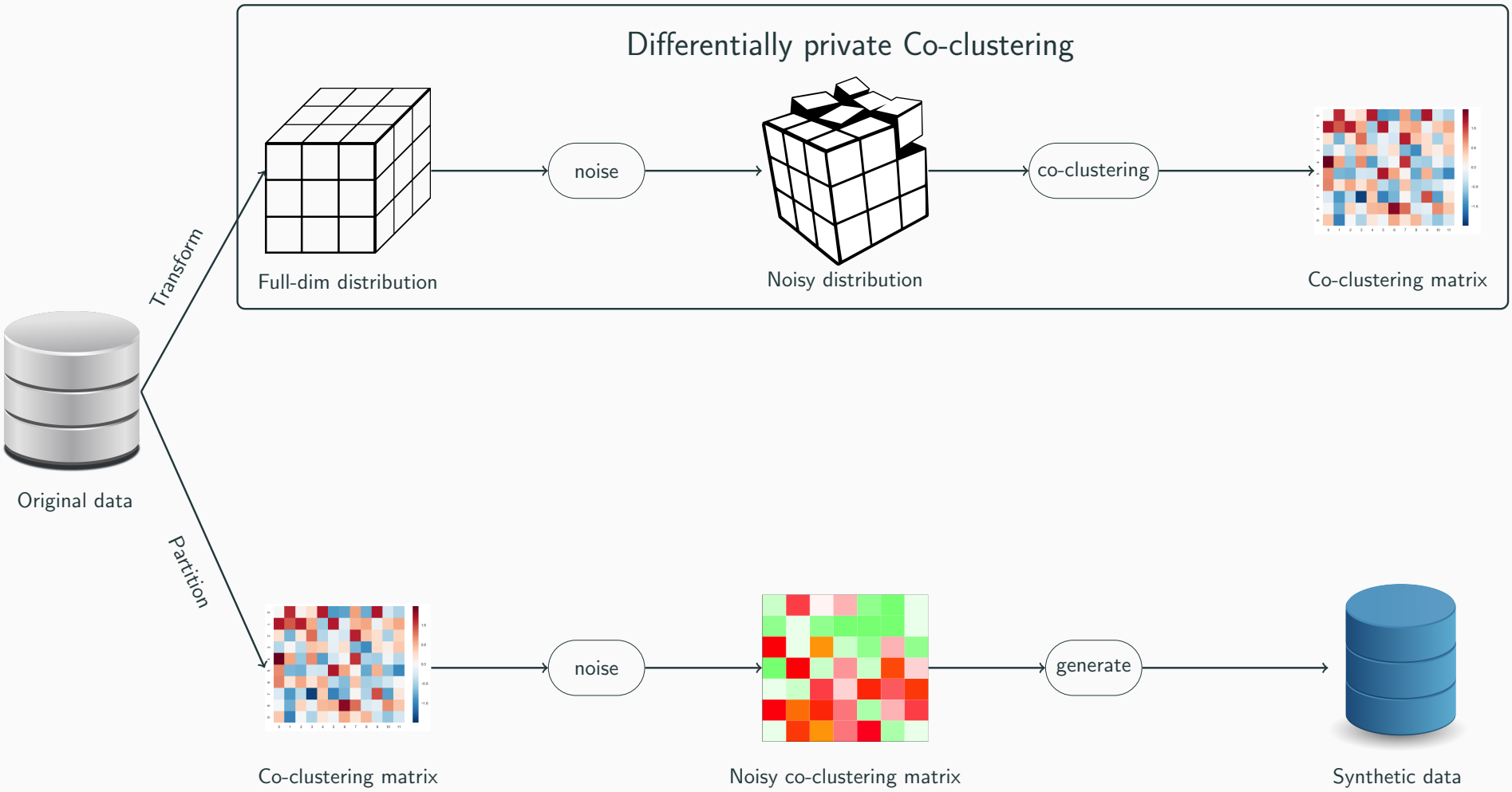
DPCocGen


Transform
Original data
Full-dim distribution
noise
Noisy distribution
co-clustering
Co-clustering matrix

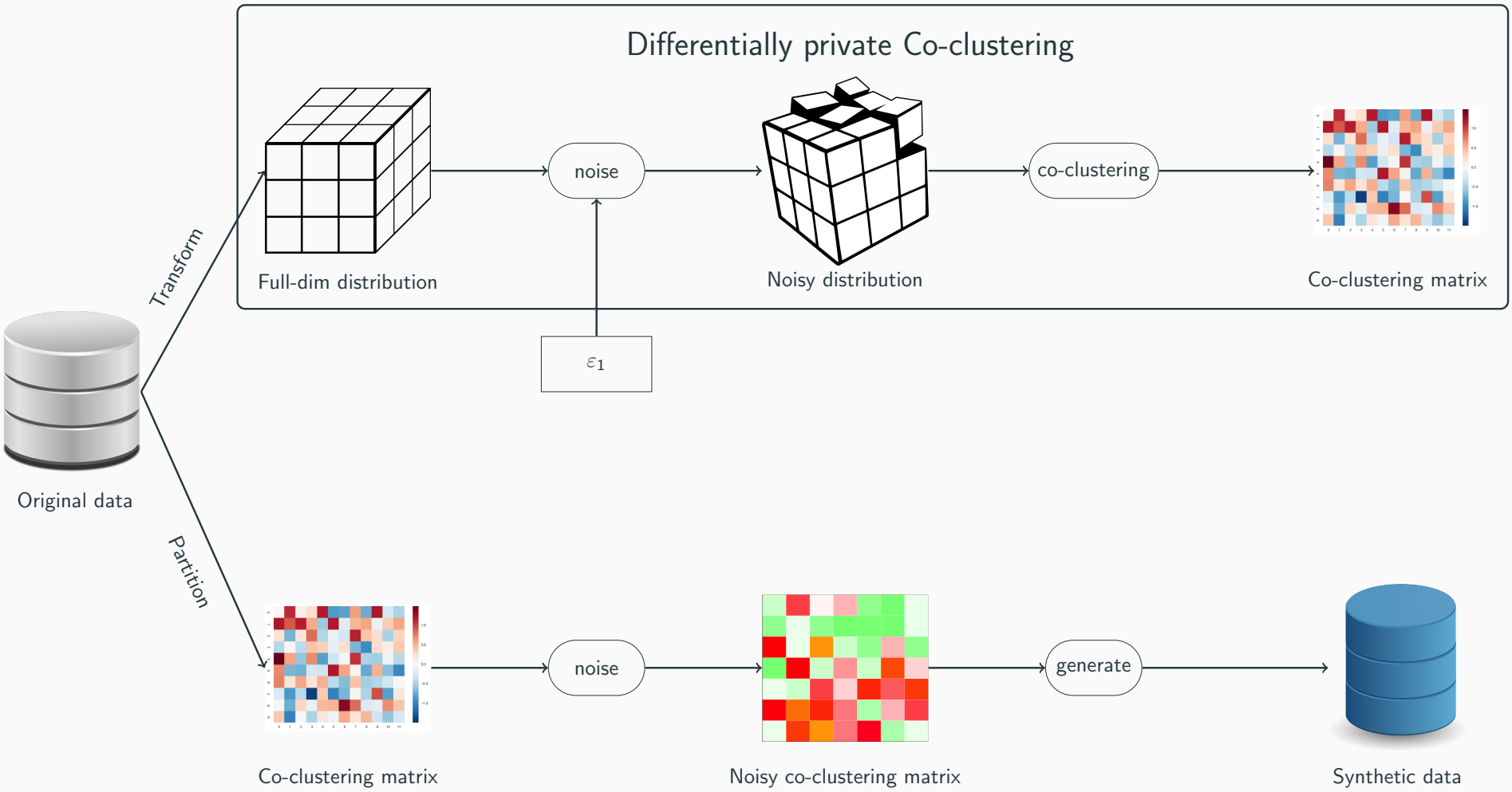
DPCocGen


Differentially private Co-clustering
noise
co-clustering
Transform
Full-dim distribution
Noisy distribution
Co-clustering matrix
Original data

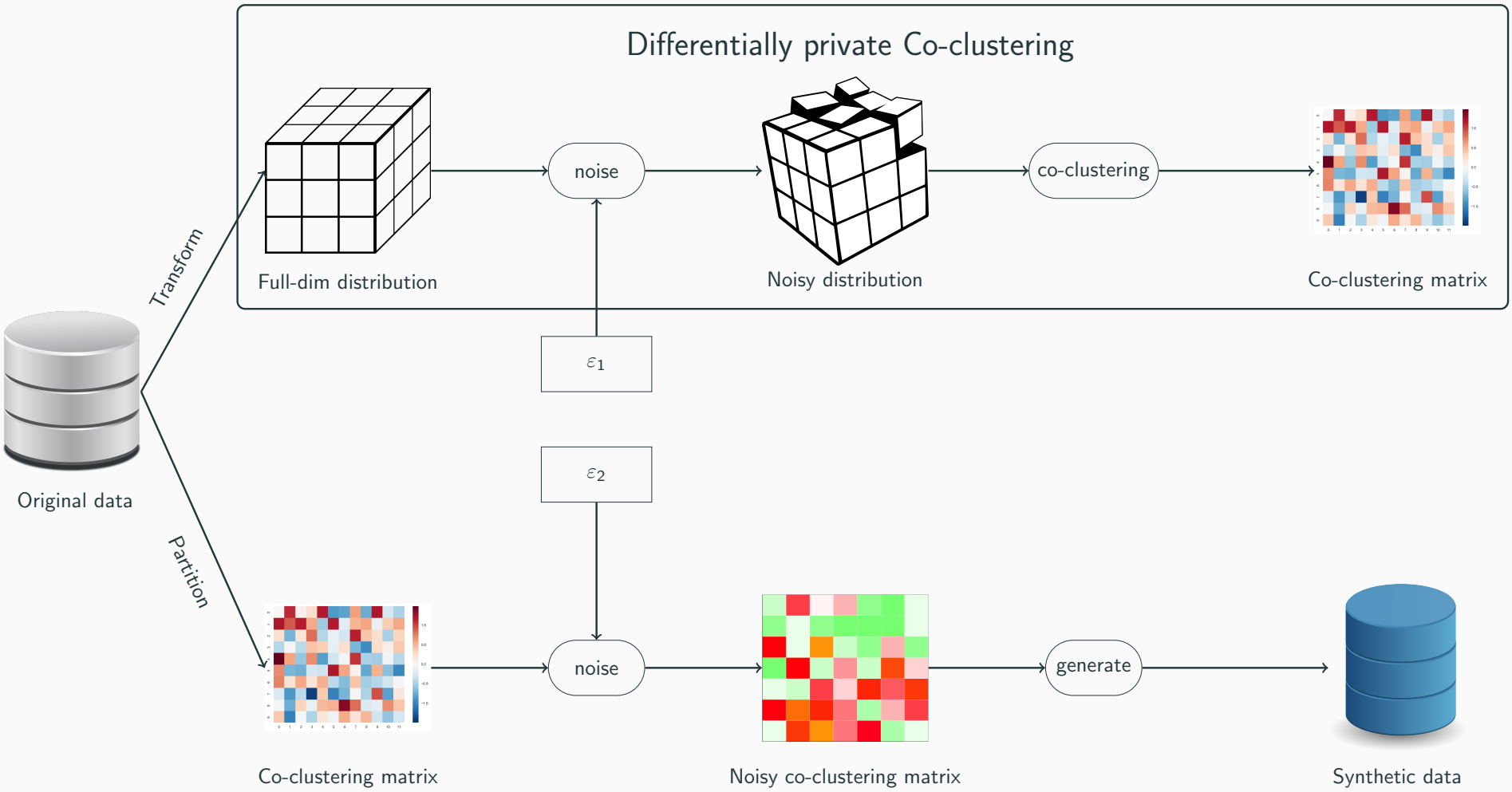
DPCocGen


Differentially private Co-clustering
Full-dim distribution
noise
Noisy distribution
co-clustering
Co-clustering matrix
Transform
Original data
Partition
Co-clustering matrix
noise
Noisy co-clustering matrix
generate
Synthetic data

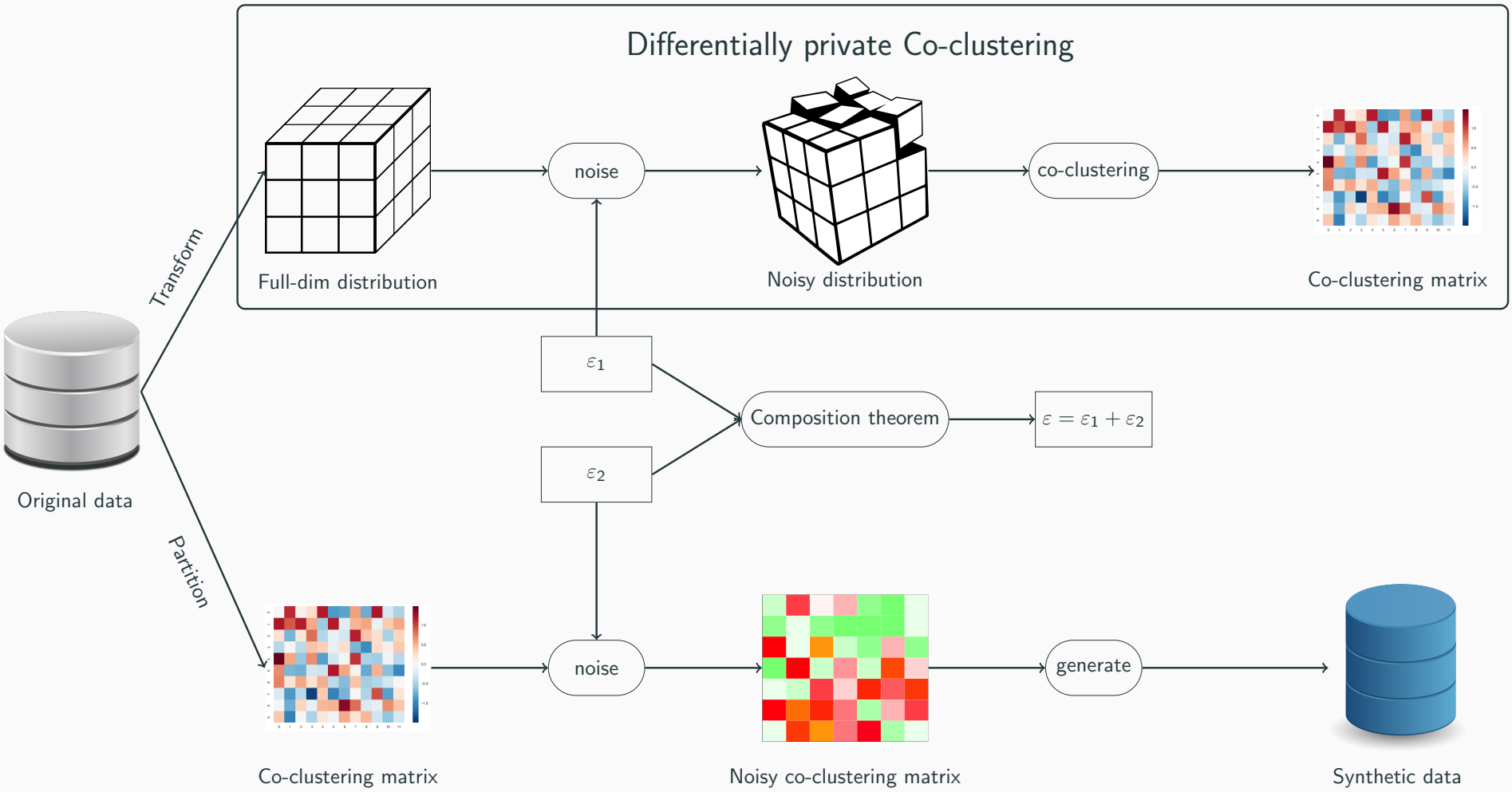
DPCocGen


Differentially private Co-clustering
Full-dim distribution
noise
Noisy distribution
co-clustering
Co-clustering matrix
ε_1
Transform
Original data
Partition
Co-clustering matrix
noise
Noisy co-clustering matrix
generate
Synthetic data

DPCocGen


Differentially private Co-clustering
Full-dim distribution
noise
Noisy distribution
co-clustering
Co-clustering matrix
Transform
Original data
ε_1
ε_2
Partition
Co-clustering matrix
noise
Noisy co-clustering matrix
generate
Synthetic data

DPCocGen


Differentially private Co-clustering
Full-dim distribution
noise
Noisy distribution
co-clustering
Co-clustering matrix
Transform
Original data
ε_1
ε_2
Composition theorem
$\varepsilon = \varepsilon_1 + \varepsilon_2$
Partition
Co-clustering matrix
noise
Noisy co-clustering matrix
generate
Synthetic data

Evaluation of DPCocGen

Evaluation

Criteria

1. Joint distribution preservation

To observe

Evaluation

Criteria

1. Joint distribution preservation
2. Relative error for random range queries

To observe

Evaluation

Criteria

1. Joint distribution preservation
2. Relative error for random range queries
3. Performance in classification with a classifier that learns from synthetic data

To observe

Evaluation

Criteria

1. Joint distribution preservation
2. Relative error for random range queries
3. Performance in classification with a classifier that learns from synthetic data

To observe

1. Impact of the privacy budget ε

Evaluation

Criteria

1. Joint distribution preservation
2. Relative error for random range queries
3. Performance in classification with a classifier that learns from synthetic data

To observe

1. Impact of the privacy budget ε
2. Impact of the aggregation level (number of cells)

Evaluation

Criteria

1. Joint distribution preservation
2. Relative error for random range queries
3. Performance in classification with a classifier that learns from synthetic data

To observe

1. Impact of the privacy budget ε
2. Impact of the aggregation level (number of cells)
3. Comparison with the base line algorithm and PrivBayes

Adult dataset

Adult

- The dataset [2] contains 48,842 instances and has 14 different attributes. The characteristics of the attributes are both numeric and nominal
- The attributes {age, workclass, education, relationship, sex} are retained
- We discretize continuous attributes into data-independent equi-width partitions

[2] UC Irvine Machine Learning Repository

Experiment: Multivariate distribution preservation

Hellinger distance

The Hellinger distance between two discrete probability distributions $P = (p_1, ..., p_k)$ and $Q = (q_1, ..., q_k)$ is given by :

$$D_{Hellinger}(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{k} (\sqrt{p_i} - \sqrt{q_i})^2}$$

Experiment

- Compute the multivariate distribution vector P of the original dataset

Experiment: Multivariate distribution preservation

Hellinger distance

The Hellinger distance between two discrete probability distributions $P = (p_1, ..., p_k)$ and $Q = (q_1, ..., q_k)$ is given by :

$$D_{Hellinger}(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{k} (\sqrt{p_i} - \sqrt{q_i})^2}$$

Experiment

- Compute the multivariate distribution vector P of the original dataset
- Compute the multivariate distribution vector Q of the synthetic data generated using `DPCocGen`

Experiment: Multivariate distribution preservation

Hellinger distance

The Hellinger distance between two discrete probability distributions $P = (p_1, ..., p_k)$ and $Q = (q_1, ..., q_k)$ is given by :

$$D_{Hellinger}(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{k} (\sqrt{p_i} - \sqrt{q_i})^2}$$

Experiment

- Compute the multivariate distribution vector P of the original dataset
- Compute the multivariate distribution vector Q of the synthetic data generated using `DPCocGen`
- Compute the multivariate distribution vector Q' of the synthetic data generated using Base line

Experiment: Multivariate distribution preservation

Hellinger distance

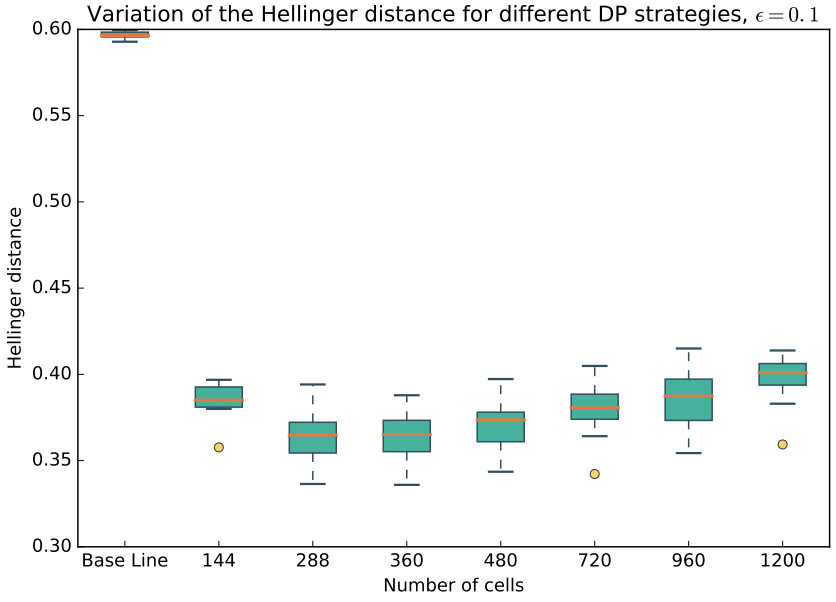
The Hellinger distance between two discrete probability distributions $P = (p_1, ..., p_k)$ and $Q = (q_1, ..., q_k)$ is given by :

$$D_{Hellinger}(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{k} (\sqrt{p_i} - \sqrt{q_i})^2}$$

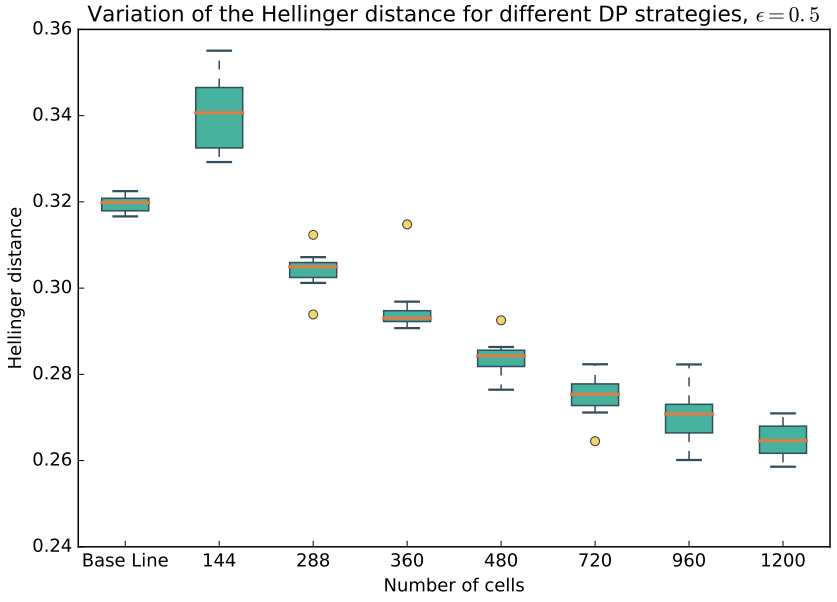
Experiment

- Compute the multivariate distribution vector P of the original dataset
- Compute the multivariate distribution vector Q of the synthetic data generated using `DPCocGen`
- Compute the multivariate distribution vector Q' of the synthetic data generated using Base line
- Compute $D_{Hellinger}(P, Q)$ and $D_{Hellinger}(P, Q')$

Results: Multivariate distribution preservation



$\varepsilon = 0.1$



$\varepsilon = 0.5$

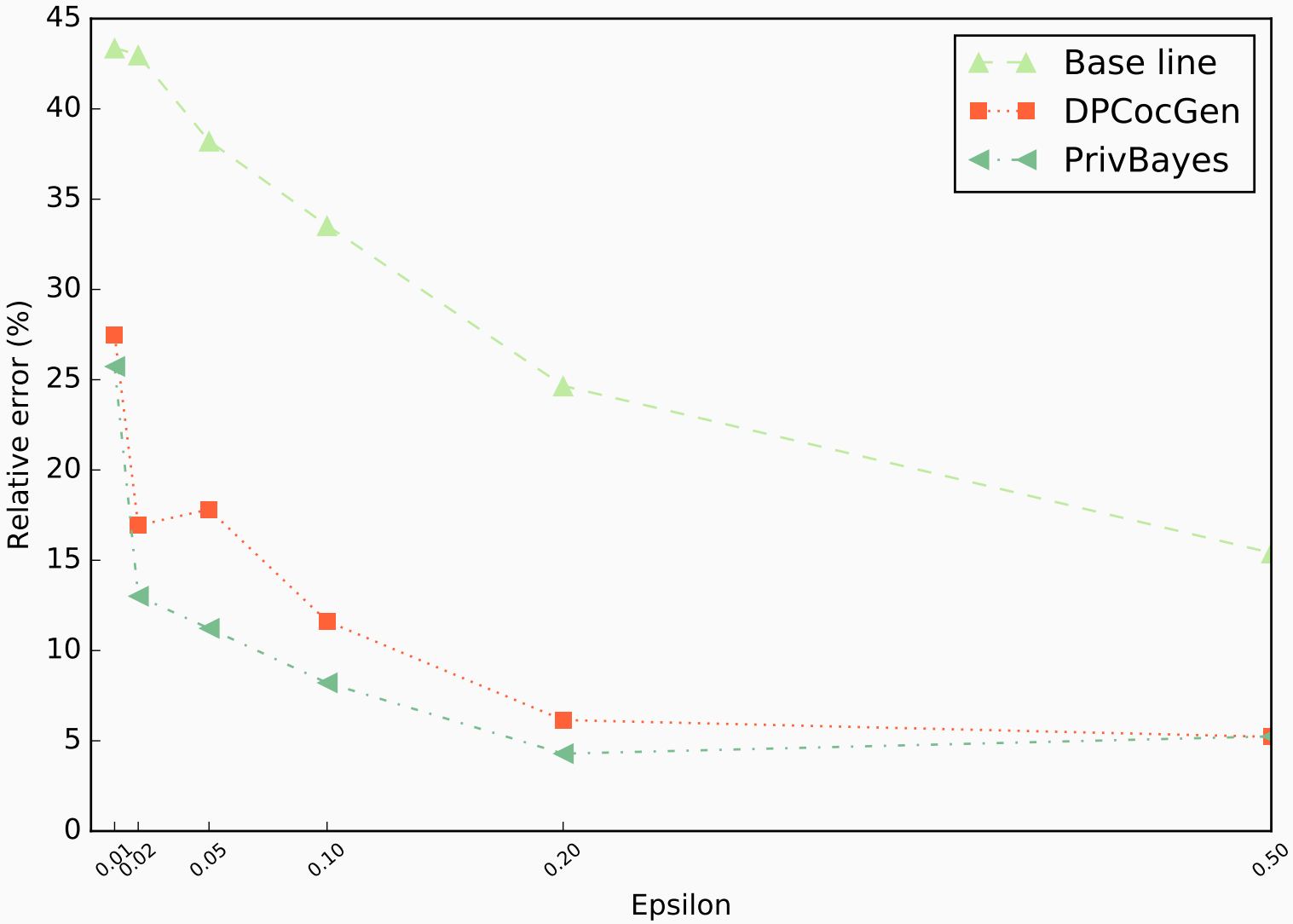
50 datasets are generated for each configuration

Experiment: Random range queries

Experiment

- Generate 100 random queries
- Compute all the queries and report their average error
- Iterate over 15 runs

Results: Random range queries



Experiment: Classification performances

Experiment

- Randomly divide the original dataset into 2 sets :
  - Training set: contains 80% of the data
  - Test set: contains 20% of the data

Experiment: Classification performances

Experiment

- Randomly divide the original dataset into 2 sets :
  - Training set: contains 80% of the data
  - Test set: contains 20% of the data
- Generate synthetic data using `DPCocGen`, Base line and PrivBayes on the Training set

Experiment: Classification performances

Experiment

- Randomly divide the original dataset into 2 sets :
  - Training set: contains 80% of the data
  - Test set: contains 20% of the data
- Generate synthetic data using `DPCocGen`, Base line and PrivBayes on the Training set
- Learn a naive Bayes classifier from the synthetic data to predict the value of the attribute *Sex*

Experiment: Classification performances

Experiment

- Randomly divide the original dataset into 2 sets :
  - Training set: contains 80% of the data
  - Test set: contains 20% of the data
- Generate synthetic data using `DPCocGen`, Base line and PrivBayes on the Training set
- Learn a naive Bayes classifier from the synthetic data to predict the value of the attribute *Sex*
- Measure classification performances of the trained models on the Test set

Classification : predict Sex

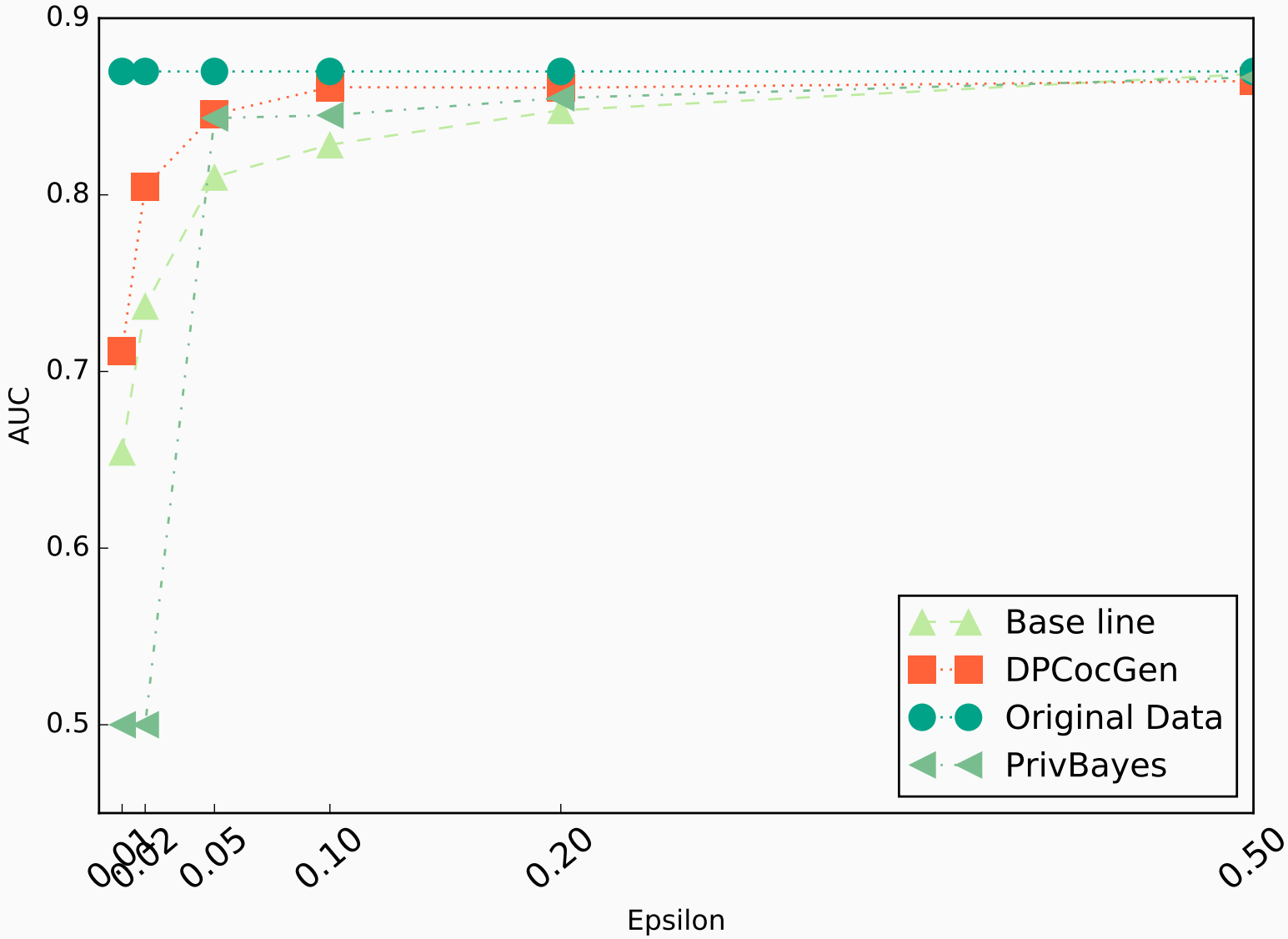


Figure 1: Average AUC, across 15 runs

Conclusion

Advantages

1. Parameter-free
2. Preserves utility

Limits

1. Limited dimension
2. Requires a discretization step

Perspectives

1. Using differentially private dimension reduction strategies, to tackle the dimension limitation

Thank you!

Cynthia Dwork.
Differential privacy.
In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer Berlin Heidelberg, 2006.

Noman Mohammed, Rui Chen, Benjamin Fung, and Philip S Yu.
Differentially private data release for data mining.
In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 493–501. ACM, 2011.

Harichandan Roy, Murat Kantarcioglu, and Latanya Sweeney.
Practical differentially private modeling of human movement data.
In *IFIP Annual Conference on Data and Applications Security and Privacy*, pages 170–178. Springer, 2016.

Yonghui Xiao, Li Xiong, Liyue Fan, and Slawomir Goryczka.
Dpcube: differentially private histogram release through multidimensional partitioning.
arXiv preprint arXiv:1202.5358, 2012.

Jun Zhang, Graham Cormode, Cecilia M Procopiuc, Divesh Srivastava, and Xiaokui Xiao.
Privbayes: Private data release via bayesian networks.
In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 1423–1434. ACM, 2014.