DPSynthesizer: Differentially Private Data Synthesizer for Privacy Preserving Data Sharing

Haoran Li, *Emory University*
Li Xiong, *Emory University*
Lifan Zhang, *Emory Univeristy*
Xiaoqian Jiang, *University of California San Diego*

Journal Title: Proceedings of the VLDB Endowment

Volume: Volume 7, Number 13

Publisher: Association for Computing Machinery | 2014, Pages 1677-1680

Type of Work: Article | Final Publisher PDF

Publisher DOI: 10.14778/2733004.2733059

Permanent URL: https://pid.emory.edu/ark:/25593/pqzft

Final published version: http://dx.doi.org/10.14778/2733004.2733059

Copyright information:

DPSynthesizer: Differentially Private Data Synthesizer for Privacy Preserving Data Sharing

Haoran Li, Li Xiong, Lifan Zhang
Math and Computer Science Department
Emory University
Atlanta, GA
hli57, lxiong, lzhan65@emory.edu

Xiaoqian Jiang
Biomedical Informatics Division
UC San Diego
La Jolla, CA
x1jiang@ucsd.edu

ABSTRACT

Differential privacy has recently emerged in private statistical data release as one of the strongest privacy guarantees. Releasing synthetic data that mimic original data with differential privacy provides a promising way for privacy preserving data sharing and analytics while providing a rigorous privacy guarantee. However, to this date there is no open-source tools that allow users to generate differentially private synthetic data, in particular, for high dimensional and large domain data. Most of the existing techniques that generate differentially private histograms or synthetic data only work well for single dimensional or low-dimensional histograms. They become problematic for high dimensional and large domain data due to increased perturbation error and computation complexity. We propose DPSynthesizer, a toolkit for differentially private data synthesization. The core of DPSynthesizer is DPCopula designed for high-dimensional and large-domain data. DPCopula computes a differentially private copula function from which synthetic data can be sampled. Copula functions are used to describe the dependence between multivariate random vectors and allow us to build the multivariate joint distribution using one-dimensional marginal distributions. DPSynthesizer also implements a set of state-of-the-art methods for building differentially private histograms, suitable for low-dimensional data, from which synthetic data can be generated. We will demonstrate the system using DPCopula as well as other methods with various data sets and show the feasibility, utility, and efficiency of various methods.

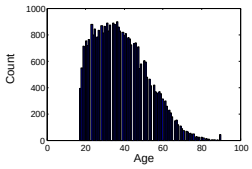
1. INTRODUCTION

Privacy preserving data analysis and publishing [2] has received considerable attention in recent years as a promising approach for sharing information while preserving data privacy. Differential privacy [5] has recently emerged as one of the strongest privacy guarantees for statistical data release. A statistical aggregation or computation is DP[1] if the outcome is formally indistinguishable when run with and without any particular record in the dataset. The level of indistinguishability is quantified by a privacy parameter ϵ. A common mechanism to achieve differential privacy is the Laplace mechanism [6] that injects calibrated noise to a statistical measure determined by the privacy parameter ϵ, and the sensitivity of the statistical measure influenced by the inclusion and exclusion of a single record in the dataset. A lower privacy parameter requires larger noise to be added and provides a higher level of privacy.

There are two main settings for differentially private data sharing. The first one is interactive setting where data users send queries to the original database through an access mechanism which returns a perturbed query answer if the allowed privacy budget on the original dataset is not exhausted based on the *composibility* of differential privacy [10]. This can be challenging in practice especially when multiple users need to pose a large number of queries for exploratory analysis. The second one is non-interactive setting where a statistical summary such as marginal or multi-dimensional histograms or a set of synthetic data that mimic the original data is publicly released in place of the original database with a given level of differential privacy and users can arbitrarily access the released data for query and analysis purposes. For example, Figure 1 shows an example dataset and a one-dimensional marginal histogram for the attribute age.

id	Age	Hours/ week	Edu	...
1	50	13	13	...
2	38	40	9	...
3	53	40	7	...
4	28	40	13	...
...	...	...	...	...

(a) Dataset



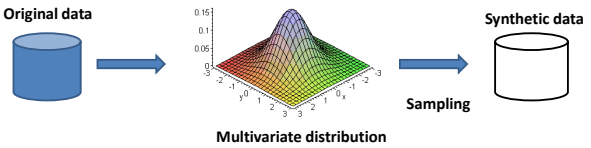
(b) Marginal Histogram for Age

Figure 1: Dataset vs. histogram illustration

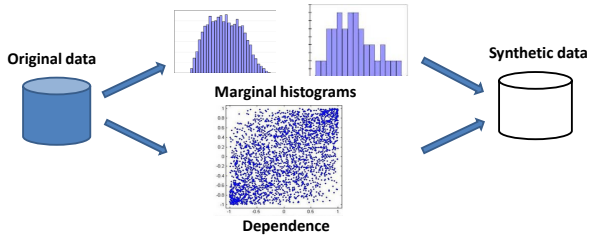
Releasing synthetic data that mimic original data with differential privacy provides a promising way for privacy preserving data sharing and analytics while providing a rigorous privacy guarantee. However, to this date there is no open-source tools that allow users to generate differentially private synthetic data, in particular, for high dimensional and large domain data. Most of the existing techniques that

[1]we shorten differentially private as DP

Proceedings of the VLDB Endowment, Vol. 7, No. 13



(a) Generate synthetic data via multivariate distribution



(b) Generate synthetic data via Copula

Figure 2: Synthetic data generation

generate differentially private histograms or synthetic data only work well for single dimensional or low-dimensional histograms. They become problematic for high dimensional and large domain data due to increased perturbation error and computation complexity. The main approaches of existing work can be illustrated by Figure 2(a) and classified into two categories: 1) parametric methods that fit the original data to a multivariate distribution and makes inferences about the parameters of the distribution (e.g. [9]). 2) non-parametric methods that learn empirical distributions from the data through histograms (e.g. [7, 12, 3, 4]). Most of these work well for single dimensional or low-order data, but become problematic for data with high dimensions and large attribute domains. This is due to the facts that:
1) The underlying distribution of the data may be unknown in many cases or different from the assumed distribution, especially for data with arbitrary margins and high dimensions, leading the synthetic data generated by the parametric methods not useful;
2) The high dimensions and large attribute domains result in a large number of histogram bins that may have skewed distributions or extremely low counts, leading to significant perturbation or estimation errors in the non-parametric histogram methods;
3) The large domain space $\prod_{i=1}^{m} |A_i|$ [2] (i.e. the number of histogram bins) incurs a high computation complexity both in time and space. For DP histogram methods that use the original histogram as inputs, it is infeasible to read all histogram bins into memory simultaneously due to memory constraints, and external algorithms need to be considered.

In this demo, we present DPSynthesizer, a toolkit for differentially private data synthesization. The core of DPSynthesizer is DPCopula, a novel differentially private data synthesization method for high dimensional and large domain data using copula functions. The system implements and extends our recent work [8] as well as several state-of-the-art histogram methods and presents several contributions.

First, DPSynthesizer implements DPCopula [8] for generating high-dimensional and large domain DP synthetic data using copula functions. Copula functions are a family of distribution functions representing the dependence structure implicit in a multivariate random vector. Intuitively, any high-dimensional data can be modeled as two parts: 1) marginal distributions of each individual dimension, and 2) the dependence among the dimensions. Copula functions have been shown to be effective for modeling high-dimensional joint distributions based on continuous marginal distributions. The key innovation of DPCopula is that it utilizes semi-parametric copula functions to separately consider the marginal histograms (non-parametric) for each single dimension and the joint dependence (parametric) among all dimensions, as shown in figure 2(b). One of the advantage of DPCopula is that it can utilize any state-of-the-art histogram method for building marginal DP histograms. In addition, DPCopula allows direct sampling for the synthetic data from the generated DP joint distribution. Although existing histogram techniques can be used to generate DP synthetic data, post-processing is required to enforce non-negative histogram counts or consistencies between counts resulting in either degraded accuracy or high computation complexity.

Second, DPSynthesizer includes a set of representative histogram methods, which can be used as a component of DPCopula for generating marginal histograms, or can be used as independent methods for generating histograms and synthetic data for low-dimensional datasets. We will demonstrate the system using various real datasets and show the feasibility, utility, and efficiency of various methods.

Finally, DPSynthesizer provides an easy-to-use web-based interface. Through the interface, users can upload their own datasets, configure parameter settings, visualize original and generated synthetic data, examine utility results, and compare different methods.

2. SYSTEM OVERVIEW

In this section, we present DPCopula, the core of DPSynthesizer, and its key steps. As seen in Figure 1(a), the input data is a relational data table while the output is a differentially private synthetic data table. DPCopula consists of several key steps: 1) estimate marginal empirical distributions via private marginal histograms, 2) estimate private dependence using gaussian copula function and the original data, 3) sample synthetic data from the marginal distributions and copula function. Below we describe each step with some technical details.

2.1 Computing DP marginal histograms

As a first step, we compute DP marginal histograms for each attribute. Several state-of-the-art techniques have been proposed for computing one-dimensional DP histograms effectively and efficiently, such as PSD [3], Privelet [11], NoiseFirst and StructureFirst [12], EFPA [1]. An important feature of DPCopula is that it can take advantage of any existing methods to compute private marginal histograms for each dimension, which can be then used to obtain empirical marginal distributions.

2.2 Computing DP dependence

We model the dependence via Gaussian copula function, a commonly used elliptical class of copula families modeling the Gaussian dependence, with its density function denoted

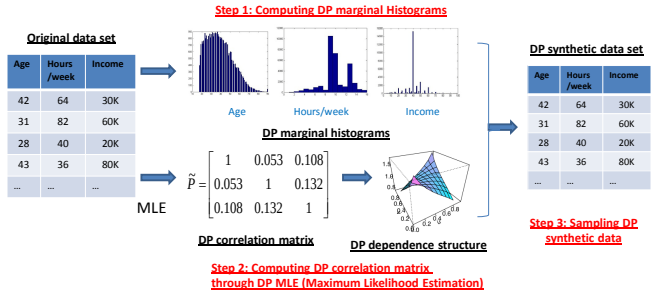
[2]We define $\prod_{i=1}^{m} |A_i|$ as the domain space of all dimensions, where $|A_i|$ is the domain size of the ith attribute and m is the number of attributes

as

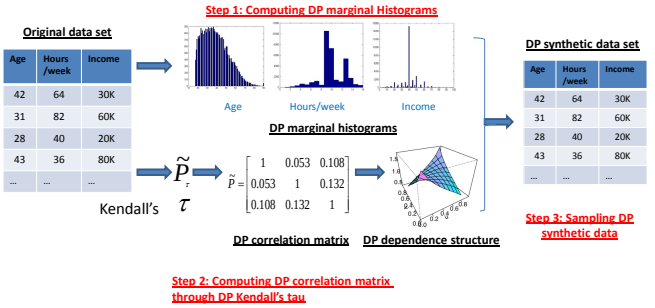
$$c_{\mathbf{P}}^{Ga} = |\mathbf{P}|^{-\frac{1}{2}} exp\left\{-\frac{1}{2}\phi^{-1}(u)^T(\mathbf{P}^{-1} - \mathbf{I})\phi^{-1}(u)\right\} \quad (1)$$

where $\mathbf{P}$ is a correlation matrix[3], $\mathbf{I}$ is the identity matrix, ϕ^{-1} is the inverse CDF of a univariate standard Gaussian distribution.

We implemented two methods, DPCopula-MLE (maximum likelihood estimation) and DPCopula-Kendall [8], to estimate the correlation matrix $\mathbf{P}$ in equation (1). An overview of DPCopula-MLE and DPCopula-Kendall is illustrated in figure 3, where the original data set contains three attributes: age, hours/week, and income, and the dependence structure is modeled by Gaussian copula.



(a) DPCopula-MLE overview



(b) DPCopula-Kendall overview

Figure 3: DPCopula techniques overview

In DPCopula-MLE, it partitions the original data D horizontally into l disjoint partitions of $\frac{n}{l}$ records each, computes a correlation matrix $\mathbf{P}_i$ $(1 \leq i \leq l)$ on each partition using MLE, and then releases the average of these estimated matrices with some small additive noise injected to each entry. The noise follows Laplace distribution which is $Lap(\frac{(\frac{m}{2})\Lambda}{l\epsilon_2})$, where Λ is the diameter of each correlation coefficient space Θ with a value of 2.

In DPCopula-Kendall, the differentially private estimator $\tilde{\mathbf{P}}$ of the general correlation matrix is estimated by calculating noisy pairwise Kendall' τ correlation coefficients matrix. From the original data vector $(\mathbf{X}_1, \ldots, \mathbf{X}_m)$, we can compute a noisy Kendall's τ coefficient of any arbitrary two attributes $\mathbf{X}_j$ and $\mathbf{X}_k$ by the standard sample Kendall's τ coefficient $\tilde{\rho}_\tau(\mathbf{X}_j, \mathbf{X}_k)$ using Laplace mechanism that guarantees differential privacy. We then construct a noisy Kendall' τ matrix $\tilde{\rho}_\tau$ with each element defined by $\tilde{\rho}_{jk}^{\tau} = \tilde{\rho}_\tau(\mathbf{X}_j, \mathbf{X}_k)$. Finally, we construct the noisy correlation matrix estimator as $\tilde{\mathbf{P}} = sin(\frac{\pi}{2}\tilde{\rho}_\tau)$ with all diagonal entries being 1.

[3]Here $\mathbf{P}$ must be a positive definite matrix to ensure that $\mathbf{P}^{-1}$ exists

2.3 Sampling DP synthetic data

Once the DP marginal histograms and DP correlation matrix are generated from the previous two steps, we can sample synthetic data. We first generate DP pseudo-copula synthetic data $(\tilde{\mathbf{T}}_1, \ldots, \tilde{\mathbf{T}}_m)$ by generating a multivariate random vector $(\tilde{\mathbf{X}}_1, \ldots, \tilde{\mathbf{X}}_m)$ following Gaussian joint distribution $\Phi(0, \tilde{\mathbf{P}})$, then transforming $(\tilde{\mathbf{X}}_1, \ldots, \tilde{\mathbf{X}}_m)$ to $(\tilde{\mathbf{T}}_1, \ldots, \tilde{\mathbf{T}}_m) \in [0,1]^{n \times m}$. Here, $\tilde{\mathbf{P}}$ is the DP correlation matrix from the second step, and $\tilde{\mathbf{T}}_j = \phi(\tilde{\mathbf{X}}_j), j = 1, \ldots, m$ and $\phi(\tilde{\mathbf{X}}_j)$ is the standard Gaussian distribution. Second, we compute DP synthetic data $\tilde{D}$ via $\tilde{D} = (\widetilde{F}_1^{-1}(\tilde{\mathbf{T}}_1), \ldots, \widetilde{F}_m^{-1}(\tilde{\mathbf{T}}_m))$, where $\widetilde{F}_j^{-1}$ is the output of the first step, the inverse of DP empirical marginal distribution function generated from the jth DP marginal histogram.

2.4 DPCopula-hybrid

Although DPCopula can model continuous attributes and discrete attributes with a large domain (e.g. attributes with the number of values no less than 10), it cannot directly handle attributes with small domains. DPSynthesizer also contains a DPCopula-hybrid method, which first partitions the original dataset based on small-domain attributes (i.e. gender) and computes the number of records for each partition in a differentially private way, then applies DPCopula-MLE or DPCopula-Kendall on each partition.

3. SYSTEM DEMONSTRATION

DPSynthesizer is a web-based application implemented with Django Framework on python. It can invoke binaries of DPCopula and other different methods. For example, current DPCopula is implemented in Matlab. The web interface is still a work in progress. A preliminary version can be accessed at http://www.mathcs.emory.edu/aims/DPSynthesizer.

Our demo mainly includes two scenarios: 1) data synthesization using DPCopula with visualization of the original, intermediate, and released DP synthetic data, 2) utility analysis and comparison of different data synthesization methods.

3.1 Data sets

We prepared two real datasets for the demonstration purposes: Brazil Census dataset (https://international.ipums.org) and US census dataset (http://www.ipums.org). The Brazil census dataset has 188,846 records after filtering out records with missing values and eight attributes: age, gender, disability, nativity, working hours per week, education, number of years residing in the current location, and annual income. We generalized the domain of income to 586. The US Census dataset has a randomly selected 100,000 records from the original 10 million records and four attributes: age, occupation, income and gender. For nominal attributes, we converted them to numeric attributes by imposing a total order on the domain of the attribute.

In order to show the impact of other factors on the utility and scalability of DPCopula and other methods in DPSynthesizer to the audience, such as distribution, dimensionality of the datasets, we also prepared synthetic datasets with 50000 records. The default attribute domain size is 1000 and each margin follows the Gaussian distribution by default.

3.2 Basic Data Synthesization Functionalities

DPSynthesizer is an easy-to-use web system that guides users to accomplish their data synthesization tasks under differential privacy. The DPSynthesizer interface allows users to select and load original datasets, to enter different parameters, to choose different types of methods according to their data needs, and to examine intermediate as well as final results. It also allows audience to visualize the original data and private released synthetic data, generate the synthetic data to a user-defined file format, and issue queries to the synthetic data for evaluating the utility.

We will start by guiding the audience through the settings of DPCopula. The users will have an opportunity to specify values for some parameters such as the overall privacy guarantee, i.e. ϵ, and the sampling rate in compute Kendall' τ correlation matrix, while these and the remaining parameters can be also automatically computed or set to default values. The system will then generate and write the released data into a file with different formats, such as ".csv". Users can select their own file path and preferred file format.

3.3 Data Visualization

DPSynthesizer provides an interface for visualization of the original and private synthetic data through histograms. For original or released synthetic data with one or two dimensions, it directly visualizes them through one or two dimensional histograms. For data with more than two dimensions, it visualizes the marginal histograms for each dimension and the correlation matrix among all dimensions. Future development of the system includes integration of visualization techniques for high-dimensional data. During the running process of DPCopula, all intermediate results, such as private marginal histograms and correlation matrix, can be also visualized. Figure 3 shows examples of the visualized marginal histograms of the attribute "Age", "Hours/week", and "Income" using the US census data, correlation matrix, and dependence structures.

3.4 Utility Analysis and Comparison

DPSynthesizer also includes the implementations of the state-of-the-art histogram methods, such as PSD (Private Spatial Decomposition), KD-hybrid methods [3], Privelet+ [11], Filter Priority (FP) with consistency checks [4], and P-HP [1]. Users can select one or more methods and compare their utility and efficiency on selected datasets and directly observe differences in results and performance.

For utility analysis, we issue random range-count queries with random query predicates on both original data and the synthesized data. The queries are defined as: Select COUNT(*) from D Where $A_1 \in I_1$ and $A_2 \in I_2$ and ... and $A_m \in I_m$. For each attribute A_i, I_i is a random interval generated from the domain of A_i.

The query accuracy is primarily measured by the relative error. For a query q, $A_{act}(q)$ is the true answer to q on the original data. $A_{noisy}(q)$ denotes the answer to q when using DP synthetic data generated from DPCopula or the DP histogram constructed by other methods. Then the relative error is defined as:

$$RE(q) = \frac{|A_{noisy}(q) - A_{act}(q)|}{max\{A_{act}(q), s\}}$$

where s is a sanity bound to mitigate the effects of queries with extremely small query answers (a commonly used evaluation method from existing literatures, e.g. [11]). While we primarily use relative error, we also use absolute error when it is more appropriate and clear to show the results for extremely sparse data, in which case, the true answers are extremely small. The absolute error is defined as $ABS(q) = |A_{noisy}(q) - A_{act}(q)|$.

Through the user interface, the audience can freely issue predicate queries using different parameter settings and observe the result. Figure 4 provides a visualization example for the error comparison result between DPCopula and other methods. The released synthetic data can be also used for learning tasks such as construction of decision tree and regression analysis. We will use classification and linear regression analysis as examples to illustrate the utility of the released synthetic data.

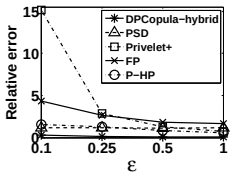


Figure 4: Comparison with other methods

Acknowledgments

This research is supported by the National Science Foundation under Grant No. 1117763.

4. REFERENCES

[1] G. Ács, C. Castelluccia, and R. Chen. Differentially private histogram publishing through lossy compression. In *ICDM*, 2012.

[2] R. C. Benjamin C.M. Fung, Ke Wang and P. S. Yu. Privacy-preserving data publishing: A survey of recent developments, ACM Computing Surveys, 2010.

[3] G. Cormode, C. M. Procopiuc, D. Srivastava, E. Shen, and T. Yu. Differentially private spatial decompositions. In *ICDE*, 2012.

[4] G. Cormode, C. M. Procopiuc, D. Srivastava, and T. T. L. Tran. Differentially private summaries for sparse data. In *ICDT*, pages 299–311, 2012.

[5] C. Dwork. Differential privacy: A survey of results. In M. Agrawal, D.-Z. Du, Z. Duan, and A. Li, editors, *TAMC*, volume 4978 of *Lecture Notes in Computer Science*, pages 1–19. Springer, 2008.

[6] C. Dwork, F. Mcsherry, K. Nissim, and A. Smith. Calibrating Noise to Sensitivity in Private Data Analysis. *Theory of Cryptography*, pages 1–20.

[7] M. Hayy, V. Rastogiz, G. Miklauy, and D. Suciu. Boosting the accuracy of differentially-private histograms through consistency. *VLDB*, 2010.

[8] H. Li, L. Xiong, and X. Jiang. Differentially private synthesization of multi-dimensional data using copula functions. In *EDBT*, 2014.

[9] A. Machanavajjhala, D. Kifer, J. Abowd, J. Gehrke, and L. Vilhuber. Privacy: Theory meets practice on the map, ICDE, 2008.

[10] McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *SIGMOD*, New York, NY, USA, 2009. ACM.

[11] X. Xiao, G. Wang, and J. Gehrke. Differential privacy via wavelet transforms. In *ICDE*, pages 225–236, 2010.

[12] J. Xu, Z. Zhang, X. Xiao, Y. Yang, and G. Yu. Differentially private histogram publication. In *ICDE*, 2012.