Journal of Machine Learning Research 25 (2024) 1-37 Submitted 11/22; Revised 10/23; Published 4/24

Differentially Private Data Release for Mixed-type Data via Latent Factor Models

Yanqing Zhang ZHANGYANQING@YNU.EDU.CN
Yunnan Key Laboratory of Statistical Modeling and Data Analysis
Yunnan University
Southwest United Graduate School
Kunming 650500, China

Qi Xu QXU6@UCI.EDU
Department of Statistics
University of California, Irvine
Irvine, CA 92697, USA

Niansheng Tang NSTANG@YNU.EDU.CN
Yunnan Key Laboratory of Statistical Modeling and Data Analysis
Yunnan University
Kunming 650500, China

Annie Qu AQU2@UCI.EDU
Department of Statistics
University of California, Irvine
Irvine, CA 92697, USA

Editor: Po-Ling Loh

Abstract

Differential privacy is a particular data privacy-preserving technology which enables synthetic data or statistical analysis results to be released with a minimum disclosure of private information from individual records. The tradeoff between privacy-preserving and utility guarantee is always a challenge for differential privacy technology, especially for synthetic data generation. In this paper, we propose a differentially private data synthesis algorithm for mixed-type data with correlation based on latent factor models. The proposed method can add a relatively small amount of noise to synthetic data under a given level of privacy protection while capturing correlation information. Moreover, the proposed algorithm can generate synthetic data preserving the same data type as mixed-type original data, which greatly improves the utility of synthetic data. The key idea of our method is to perturb the factor matrix and factor loading matrix to construct a synthetic data generation model, and to utilize link functions with privacy protection to ensure consistency of synthetic data type with original data. The proposed method can generate privacy-preserving synthetic data at low computation cost even when the original data is high-dimensional. In theory, we establish differentially private properties of the proposed method. Our numerical studies also demonstrate superb performance of the proposed method on the utility guarantee of the statistical analysis based on privacy-preserved synthetic data.

Keywords: Correlation, Factor model, Link function, Synthetic data

1. Introduction

With the rapid development of multimodal data collection in knowledge discovery, there is a growing need for sharing datasets for reproducibility purposes and public usage. However, datasets could contain confidential or sensitive information of individuals, for example, information on personal health, income, and racial or ethnic origin. Moreover, datasets could have highly correlated attributes and mixed-data types containing continuous, ordinal and nominal data. Directly sharing such datasets might violate individual privacy. Therefore, protecting data privacy while ensuring data utility has become essential for sharing datasets. Differentially private synthetic data plays an important role in generating new datasets for sharing the same statistical properties as the original data, as well as preserving individual privacy of the original data.

Specifically, differential privacy (DP) (Dwork et al., 2006) is a notion that quantifies the privacy of a system using a solid mathematical formulation, and provides a probability definition of the degree of privacy loss in measuring the privacy guarantee of an algorithm. There are different variants of the DP including approximate DP (Dwork et al., 2006; Cai et al., 2021; Liu et al., 2021), local DP (Evfimievski et al., 2003; Duchi et al., 2013; Rohde and Steinberger, 2020), random DP (Hall et al., 2013), Renyi DP (Mironov, 2017), and Gaussian DP (Dong et al., 2019; Zheng et al., 2021). Moreover, the principle of a DP algorithm is to reject random noise to ensure privacy protection in that the released data information does not change much if one individual in the dataset changes. The DP algorithms have been deployed at large scales in practice by organizations such as Apple, Google and the U.S. Census Bureau. Usage of DP mainly focuses on releasing certain data analysis results with privacy protection (Dwork and Smith, 2010; Duchi et al., 2018; Bu et al., 2020; Avella-Medina, 2021; Awan and Slavković, 2021), and releasing privacy-preserving synthetic data (Hardt and Rothblum, 2010; Ping et al., 2017; Abay et al., 2018; Mckenna et al., 2019; McKenna et al., 2021). The former has to pre-specify certain analyses and thus makes data sharing more restrictive; whereas the latter offers privacy-preserving data releasing so that the releasing dataset grants analysts more freedom to perform any analysis with their own methods and models. One big challenge of privacy-preserving data release is the balance between differential privacy and utility guarantee.

Many techniques for differentially private data release have been developed, including probabilistic sampling-based generation methods (Li et al., 2014; Chen et al., 2015; Zhang et al., 2017), generative adversarial networks algorithms (Yoon et al., 2019; Acs et al., 2019; Chen et al., 2020), data release methods based on projection or transformers (Zhou et al., 2009; Xiao et al., 2011; Blocki et al., 2012; Xu et al., 2017; Upadhyay, 2018; Arora et al., 2018; Chanyaswad et al., 2019; Gondara and Wang, 2020), and synthetic data algorithms based on Bayesian networks (Ping et al., 2017; Zhang et al., 2017; Bao et al., 2021). However, these methods treat original data as continuous data regardless of the original data type, and only guarantee the generation of continuous synthetic data. Consequently, the relative consistency of data type between synthetic data and original data is ignored. This could affect the utility of synthetic data in downstream data analyses, data interpretation and machine learning algorithms.

Preserving the same types of the release data as the original data is quite essential in data curation and analyses. Existing literature on DP synthetic data methods incorporating consistency of the data type of release data include Jiang et al. (2013) based on principle components analysis (PCA), Frigerio et al. (2019) and Tantipongpipat et al. (2021) based on the generative adversarial networks algorithm, Mckenna et al. (2019) and McKenna et al. (2021) based on discrete graphical models, and Bi and Shen (2022) and Shen et al. (2022) using inverse cumulative distribution functions of continuous and discrete variables to generate privatized data. Jiang et al. (2013), Frigerio et al. (2019) and Tantipongpipat et al. (2021) used one-hot encoding for discrete variables; but cannot incorporate the correlation dependency among categorical variables, which could also increase the dimension of the input data. Furthermore, the methods based on discrete graphical models such as Mckenna et al. (2019) and McKenna et al. (2021), require that all attributes in the original data are discrete and only deal with discrete variables, although the released data can preserve discrete type similar to the original data. In addition, the construction of inverse joint cumulative distribution functions in Bi and Shen (2022) and Shen et al. (2022) could be very complex for high-dimensional correlated data.

In this paper, we build a novel differentially private data release algorithm for mixed-type data based on a latent factor model. The proposed method utilizes a Laplace mechanism to perturb the factor matrix and loading loading matrix for synthetic datasets while achieving specified privacy requirements. Specifically, we transform categorical data from the original data to be continuous data via certain link functions before constructing the latent factor model. Then, we assign a privacy budget to factor matrix and eigenvectors associated with the original data information. Furthermore, through the Laplace mechanism, we obtain a new privacy-preserving eigenvectors matrix via adding weighted noise and a perturbed factor matrix. Through the latent factor model we construct a synthetic data generation model. We assign a privacy budget to construct reverse transformations of link functions with privacy protection, and obtain synthetic data with the same data type as the original data through such reverse transformations. In theory, we show that the proposed algorithm achieves the differential privacy requirement, and we establish the upper bound of differences between synthetic data and original data.

The proposed method has three significant contributions. First, the main advantage of the proposed method due to the latent factor model is its capability of preserving the main information of correlated variables and generating high-utility synthetic data without leaking sensitive information. In fact, achieving privacy protection while preserving the correlated information of original data is critically important. For example, an individual with highly correlated features is particularly at risk of being identified. Second, the proposed method can preserve the original data type, that is, the generated synthetic data has the same data type as the original data. This could be essential in the process of data curation and data release. Our method is to preserve the data structure of the original data without losing essential information, and to preserve the correlated information among the categorical variables. Third, the proposed method only requires adding a small amount of noise to ensure the privacy guarantee of original data. Since the factor matrix is a projection under lower dimensional space, our perturbation on the factor matrix controls the noise intensity added to the data matrix to achieve noise reduction of the synthetic data. Thus, our method can maintain most of the utility of the release data.

The remainder of the paper is organized as follows. Section 2 introduces the notation and background on differential privacy. Section 3 presents the proposed method and theoretical properties. Section 4 presents simulation studies to assess the performance of the proposed approach. In Section 5, we apply the proposed method to real datasets. Concluding remarks and discussion are provided in Section 6.

2. Background of Differential Privacy

In this section, we first provide some preliminaries on differential privacy. Differential privacy is proposed for publicly sharing information of a dataset while protecting every individual's information in the dataset. Consider a set of data points $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \in \mathcal{X}^n$, where $\mathcal{X} = \mathbb{R}^p$ and $\boldsymbol{x}_i \in \mathbb{R}^p (i = 1, \ldots, n)$ is the record of an individual. We first give the definition of ϵ-differential privacy.

Definition 1 ***(ϵ-differential privacy)*** *A randomized algorithm $\mathcal{M} : \mathcal{X}^n \to \mathbb{T}$ satisfies ϵ-differential privacy if $\Pr\{\mathcal{M}(\boldsymbol{X}) \in \boldsymbol{D}\} \leq e^{\epsilon} \Pr\{\mathcal{M}(\boldsymbol{X}') \in \boldsymbol{D}\}$ for all sets of possible outputs $\boldsymbol{D} \in \mathbb{T}$ and all inputs $\boldsymbol{X} \in \mathcal{X}^n$ and $\boldsymbol{X}' \in \mathcal{X}^n$ differing in a single record.*

Definition 1 provides a description of probability as to whether an algorithm can protect individual privacy, that is, whether including or excluding a particular subject sample in the dataset could change the probability of a particular outcome. The term $\epsilon > 0$ is called the privacy budget and controls the amount of output difference from an algorithm between two adjacent databases. This also captures lots of privacy when an algorithm is implemented to databases. A smaller ϵ ensures less privacy loss and corresponds to higher privacy protection. The ϵ-differential privacy has the following important properties.

Proposition 2 ***(Composition)*** *Let $\mathcal{M}_i : \mathcal{X}^n \to \mathbb{T}_i$ be an ϵ_i-differentially private algorithm for $i = 1, 2, \ldots, k$. If $\mathcal{M}_{[k]} : \mathcal{X}^n \to \prod_{i=1}^{k} \mathbb{T}_i$ is defined as $\mathcal{M}_{[k]}(x) = (\mathcal{M}_1(x), \ldots, \mathcal{M}_k(x))$, then $\mathcal{M}_{[k]}$ is $\sum_{i=1}^{k} \epsilon_i$-differentially private.*

Proposition 2 provides the composition property of differential privacy, that is, the joint distribution of the outputs from differentially private algorithms also satisfies differential privacy. Based on this composition property, we can divide the budget ϵ over sequential algorithms to ensure that these algorithms are ϵ_i-differentially private, respectively. Understanding the behavior of differentially private algorithms under the composition property allows most of the design and analysis of complex differentially private algorithms to preserve simpler differentially private building blocks.

Proposition 3 ***(Post-Processing)*** *Let $\mathcal{M} : \mathcal{X}^n \to \mathbb{T}$ be a randomized algorithm that is ϵ-differentially private. Let $f : \mathbb{T} \to \mathbb{T}'$ be an arbitrary randomized mapping. Then $f \circ \mathcal{M} : \mathcal{X}^n \to \mathbb{T}'$ is ϵ-differentially private.*

Proposition 3 implies that differential privacy is not sensitive to post-processing. That is, without additional knowledge about the private data, one cannot derive any function of an output of a differentially private algorithm $\mathcal{M}$ to break the differential privacy constraint or increase privacy loss. Due to the post-processing property of differential privacy, we can design a complex differentially private algorithm based on some differentially private mechanisms. Classical mechanisms for designing differentially private algorithms include the

Laplace mechanism (Dwork et al., 2006), the Gaussian mechanism (Dwork and Roth, 2014) and the exponential mechanism (McSherry and Talwar, 2007). The Laplace mechanism is defined as follows.

Definition 4 ***(Laplace Mechanism)*** *For any function $f : \mathcal{X}^n \to \mathbb{R}^k$, the Laplace mechanism is defined as $\mathcal{M}(f(x), \epsilon) = f(x) + (e_1, \cdots, e_k)$, where e_i are i.i.d. random variables from a Laplace distribution with a zero location parameter and a scale parameter $\Delta f/\epsilon$, denoted as Lap$(\Delta f/\epsilon)$, in which Δf is the ℓ_1-sensitivity of $f(\cdot)$.*

The ℓ_1-sensitivity of $f(\cdot)$ is $\Delta f = \max_{|x-x'|_1=1} \|f(x) - f(x')\|_1$, where $|x - x'|_1 = 1$ denotes that x and x' differ in at most one element, and $\|f(x) - f(x')\|_1$ denotes the ℓ_1 norm of $f(x) - f(x')$. The ℓ_1-sensitivity of a function captures the magnitude by which a single individual's data can change the function in the worst case, requiring us to introduce uncertainty in the response in order to mask the participation of single individuals. Based on the framework of Dwork et al. (2006), the Laplace mechanism satisfies the ε-differential privacy restraint.

3. Proposed Method

This section describes the proposed differentially private synthetic data generation approach based on the factor model and the Laplace mechanism. We first introduce synthetic data generation approach via factor models. Then, we propose the differentially private synthetic data generation approach for mixed-type original data. In addition, we present the privacy guarantee and utility guarantee of the proposed method.

3.1 Synthetic Data via Factor Models

A factor model is an effective way of extracting information from an original dataset. In a linear factor model, the original dataset can be represented as a linear combination of a set of independent latent factors and idiosyncratic noise, where the linear combination has nearly the same correlations as the original dataset. Specifically, we consider an original dataset $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)^\top \in \mathbb{R}^{n\times p}$, where $\boldsymbol{x}_i \in \mathbb{R}^p$ is the i-th sample and correlations exist among the p-dimensional variables. We define a linear factor model based on the original dataset as follows:

$$\boldsymbol{X} = \mathbf{W}\boldsymbol{\Lambda}^\top + \mathbf{E}, \tag{1}$$

where $\mathbf{W} \in \mathbb{R}^{n\times r}$ is a matrix of latent factors, $\boldsymbol{\Lambda} \in \mathbb{R}^{p\times r}$ is a factor loading matrix, r is the number of factors, and $\mathbf{E} \in \mathbb{R}^{n\times p}$ is a random error matrix. Here, each column of $\mathbf{W}$ is an independent latent factor, and the factor loading matrix $\boldsymbol{\Lambda}$ captures the relationship of the original data and latent factors. One advantage of the factor model is summarizing the information of the original data into low-dimensional latent factors. For a fixed number of factors, once obtaining the estimators $\widehat{\mathbf{W}}$ and $\widehat{\boldsymbol{\Lambda}}$ of $\mathbf{W}$ and $\boldsymbol{\Lambda}$ in model (1), we can then generate synthetic data approximating the original data through the equation $\widehat{\boldsymbol{X}} = \widehat{\mathbf{W}}\widehat{\boldsymbol{\Lambda}}^\top$.

The key to generating synthetic data via the above low-rank approximation is to estimate $\mathbf{W}$ and $\boldsymbol{\Lambda}$. Generally, they can be estimated by minimizing the following objective function:

$$Q(\boldsymbol{\Lambda}, \mathbf{W}) = \sum_{i=1}^{n} \sum_{j=1}^{p} (x_{ij} - \mathbf{w}_i^\top \boldsymbol{\lambda}_j)^2 = \text{ tr } \left\{ (\boldsymbol{X} - \mathbf{W}\boldsymbol{\Lambda}^\top)^\top (\boldsymbol{X} - \mathbf{W}\boldsymbol{\Lambda}^\top) \right\}. \tag{2}$$

For the identification of $\mathbf{W}$ and $\boldsymbol{\Lambda}$, normalization restrictions on $\mathbf{W}$ and $\boldsymbol{\Lambda}$ are needed in optimizing (2). Specifically, based on the framework of Bai (2003), we impose the requirements that $\boldsymbol{\Lambda}^\top \boldsymbol{\Lambda} = \mathbf{I}_r$ and $\mathbf{W}^\top \mathbf{W}$ is diagonal. Through normalization $\boldsymbol{\Lambda}^\top \boldsymbol{\Lambda} = \mathbf{I}_r$ and concentrating $\mathbf{W}$, solving (2) is identical to maximizing $\text{tr}\{\boldsymbol{\Lambda}^\top (\boldsymbol{X}^\top \boldsymbol{X}) \boldsymbol{\Lambda}\}$. The solution is the estimated factor loading matrix $\widehat{\boldsymbol{\Lambda}}$ where the j-th column of $\widehat{\boldsymbol{\Lambda}}$ is the eigenvector $\boldsymbol{\mu}_j$ corresponding to the j-th largest eigenvalue ν_j of the matrix $\boldsymbol{X}^\top \boldsymbol{X}$, that is, $\widehat{\boldsymbol{\Lambda}} = \boldsymbol{V} = (\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \cdots, \boldsymbol{\mu}_r)$. The corresponding factor matrix estimator $\widehat{\mathbf{W}} = \boldsymbol{X}\widehat{\boldsymbol{\Lambda}}(\widehat{\boldsymbol{\Lambda}}^\top \widehat{\boldsymbol{\Lambda}})^{-1} = \boldsymbol{X}\widehat{\boldsymbol{\Lambda}}$. Thus, synthetic data can be generated via $\widehat{\boldsymbol{X}} = \widehat{\mathbf{W}}\widehat{\boldsymbol{\Lambda}}^\top = \boldsymbol{X}\boldsymbol{V}\boldsymbol{V}^\top$ to resemble the original data.

The above factor model transforms the correlated variables of original data to linearly independent factor variables, and all the essential information from the original data is captured while the dimensionality of the original data is reduced. Based on eigenvalue decomposition, the amount of information of original data is measured by sample covariance $\boldsymbol{X}^\top \boldsymbol{X}/n$, where eigenvectors reflect the directions of the spread of original data, and eigenvalues are the magnitudes of the spreads of the corresponding direction. Therefore, a larger ν_j indicates that the corresponding eigenvector represents more variations and information from the original dataset. Consequently, the eigenvectors corresponding to the first r largest eigenvalues represent most of the information of the matrix $\boldsymbol{X}^\top \boldsymbol{X}$.

Based on the above estimation, we can select a proper number of eigenvalues to construct the estimator $\widetilde{\boldsymbol{\Lambda}}$ storing most of the information of the original data. There are many methods for determining the number of factors, for example, cumulative information ratio, panel C_p information criteria (Bai and Ng, 2002), empirical distribution of eigenvalues (Onatski, 2010), and through maximizing the ratios of two adjacent eigenvalues (Ahn and Horenstein, 2013; Wu, 2016; Xia et al., 2017). We set the number of factors based on the cumulative information ratio, that is,

$$r(c) = \arg \min_{1 \le k < q} \left\{ k : \frac{\sum_{j=1}^{k} \nu_j}{\sum_{j=1}^{q} \nu_j} > c \right\}, \tag{3}$$

where $q = \min\{n, p\}$ and $c \in (0, 1)$ is a given threshold, e.g., $c = 0.8$ or 0.9.

Although we can release synthetic data via the above factor model, such synthetic data do not necessarily possess a low level of disclosure risk. Privacy leakage can still occur since the synthetic data $\widehat{\boldsymbol{X}} = \boldsymbol{X}\boldsymbol{V}\boldsymbol{V}^\top$ is not random and could be still sensitive to the original data. The potential privacy leakage motivates the design of the data releasing algorithm with differential privacy guarantee in the following subsection.

3.2 Differentially Private Synthetic Data via Factor Models

We consider a data matrix $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\top \in \mathbb{R}^{n \times p}$ and assume that $\boldsymbol{x}_i$ satisfies $\|\boldsymbol{x}_i\|_2 \le 1$. Building a factor model (1) is through calculating eigenvectors of the matrix $\boldsymbol{X}^\top \boldsymbol{X}$.

Before constructing the differentially private algorithm, we first establish the sensitivities of the eigenvectors.

Lemma 5 *Denote $\boldsymbol{\mu}_i : \mathbb{R}^{n\times p} \to \mathbb{R}^p$ for $i = 1, 2, \ldots, q$ as the i-th eigenvector corresponding to the i-th largest eigenvalue of the matrix $\boldsymbol{X}^\top \boldsymbol{X}$, where $\boldsymbol{X} \in \mathbb{R}^{n\times p}$. Then the ℓ_1-sensitivity of the eigenvector $\boldsymbol{\mu}_i$ satisfies*

$$\Delta\boldsymbol{\mu}_i = \max_{\boldsymbol{X}':|\boldsymbol{X}-\boldsymbol{X}'|_1=1} \|\boldsymbol{\mu}_i(\boldsymbol{X}) - \boldsymbol{\mu}_i(\boldsymbol{X}')\|_1 \leq 2\sqrt{p},$$

where $|\boldsymbol{X} - \boldsymbol{X}'|_1 = 1$ denotes that $\boldsymbol{X}$ and $\boldsymbol{X}'$ contain at most one different record.

Lemma 5 provides the sensitivity of each eigenvector of the matrix $\boldsymbol{X}^\top\boldsymbol{X}$. Although eigenvectors corresponding to different eigenvalues represent varying levels of variation and information of the original data, the sensitivities of all eigenvectors have a uniform upper bound which does not depend on the magnitude of eigenvalues.

Lemma 6 *Denote $\mathbf{W}(\boldsymbol{X}) = \boldsymbol{X}\mathbf{T}$ for any fixed matrix $\mathbf{T} = (\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_r)$, where $\boldsymbol{\mu}_k$ is a fixed unit vector for $k = 1, \ldots, r$ and $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\top \in \mathbb{R}^{n\times p}$ satisfies $\|\boldsymbol{x}_i\|_2 \leq 1$ for $i = 1, \cdots, n$. Then the ℓ_1-sensitivity of the vector function $\mathbf{W}(\cdot)$ satisfies*

$$\Delta\mathbf{W} = \max_{\boldsymbol{X}':|\boldsymbol{X}-\boldsymbol{X}'|_1=1} \|\mathbf{W}(\boldsymbol{X}) - \mathbf{W}(\boldsymbol{X}')\|_1 \leq 2r,$$

where $|\boldsymbol{X} - \boldsymbol{X}'|_1 = 1$ denotes that $\boldsymbol{X}$ and $\boldsymbol{X}'$ contain at most one different record, and the ℓ_1-norm of a matrix $\boldsymbol{A} \in \mathbb{R}^{n\times p}$ is $\|\boldsymbol{A}\|_1 = \sum_{i=1}^n \sum_{j=1}^p |a_{ij}|$.

Lemma 6 provides the sensitivity of the function $\mathbf{W}(\boldsymbol{X})$ given any fixed matrix with unit column vector. When the unit column vector $\boldsymbol{\mu}_k$ is any one of eigenvectors of the matrix $\boldsymbol{X}^\top\boldsymbol{X}$, the matrix $\mathbf{W}(\boldsymbol{X})$ is an estimated factor matrix in the model (1).

Based on the above sensitivities, we design noise-adding methods in the factor matrix and loading matrix to obtain synthetic data with differential privacy. Specifically, we construct a perturbed factor model for synthetic data generation as follows:

$$\widetilde{\boldsymbol{X}} = \widetilde{\mathbf{W}}\widetilde{\boldsymbol{\Lambda}}^\top := (\boldsymbol{X} \cdot \boldsymbol{g}(\boldsymbol{V} + \mathbf{B}) + \boldsymbol{C}) \cdot \boldsymbol{g}(\boldsymbol{V} + \mathbf{B})^\top, \tag{4}$$

where $\widetilde{\boldsymbol{\Lambda}} = \boldsymbol{g}(\boldsymbol{V} + \mathbf{B})$ is the perturbed loading matrix, $\widetilde{\mathbf{W}} = (\boldsymbol{X} \cdot \boldsymbol{g}(\boldsymbol{V} + \mathbf{B}) + \boldsymbol{C})$ is the perturbed factor matrix, $\boldsymbol{V}$ is the eigenvector matrix corresponding to the first r largest eigenvalues of the matrix $\boldsymbol{X}^\top\boldsymbol{X}$ in a decreasing order, $\mathbf{B}$ is a $p \times r$ noise matrix, $\boldsymbol{C}$ is a $n \times r$ noise matrix, and $\boldsymbol{g}(\boldsymbol{V} + \mathbf{B})$ is the top r left-singular-vector matrix of the matrix $\boldsymbol{V} + \mathbf{B}$ for obtaining a unit orthonormal loading matrix. Based on Definition 4 and Lemma 5, we construct the noise matrix $\mathbf{B}$ with each entry of the i-th column from $Lap(2\sqrt{p}/\epsilon_{1i})$ for $i = 1, \ldots, n$, where $\epsilon_{1i} = \omega_i\epsilon_1$, $\omega_i > 0$ and $\sum_{i=1}^r \omega_i = 1$. Thus, the matrix $\boldsymbol{V} + \mathbf{B}$ is a perturbed eigenvector matrix. To construct a perturbed loading matrix satisfying the restriction of $\boldsymbol{\Lambda}^\top\boldsymbol{\Lambda} = \mathbf{I}_r$, we consider singular value decomposition of the perturbed eigenvector matrix. That is, $\boldsymbol{V} + \mathbf{B} = \boldsymbol{SQR}$, where $\boldsymbol{Q}$ is a $p \times r$ rectangular diagonal matrix with the singular values of $\boldsymbol{V} + \mathbf{B}$ in decreasing order on the diagonal, $\boldsymbol{R}$ is a $r \times r$ unit orthogonal matrix with each column being the corresponding right-singular vector, and

$\boldsymbol{S}$ is a $p \times p$ unit orthogonal matrix with each column being the corresponding left-singular vector. Then the perturbed loading matrix is $\boldsymbol{g}(\boldsymbol{V} + \mathbf{B}) = \boldsymbol{S}_r$, where $\boldsymbol{S}_r$ is a $p \times r$ unit orthogonal matrix with the columns being the top r left-singular vectors. Note that the orthonormalization of the perturbed loading matrix is necessary to construct a perturbed factor matrix based on Lemma 6. Based on Definition 4 and Lemma 6, we construct the noise matrix $\boldsymbol{C}$ with each entry from $Lap(2r/\epsilon_2)$. Thus, $\widetilde{\mathbf{W}} = \boldsymbol{X} \cdot \boldsymbol{g}(\boldsymbol{V} + \mathbf{B}) + \boldsymbol{C}$ is a perturbed factor matrix. Based on Definition 1, we can show that the synthetic data $\widetilde{\boldsymbol{X}}$ generated from the model (4) satisfies the differential privacy requirement indicated in Theorem 10. The details of the synthetic data generation algorithm based on model (4) are summarized as follows.

Algorithm 1 Differentially private data-releasing algorithm based on factor models

```
Input:
    Original data $\boldsymbol{X}$ ($n \times p$), privacy budget $\epsilon = \epsilon_1 + \epsilon_2$ and the number of factors $r$;
Output:
    Synthesized data $\widetilde{\boldsymbol{X}}$;
1: Calculate matrix $\boldsymbol{A} = \boldsymbol{X}^\top \boldsymbol{X}$ and $\boldsymbol{V} = (\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \cdots, \boldsymbol{\mu}_r)$ with the columns being the eigenvectors corresponding to the first $r$ largest eigenvalues $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_r)^\top$ of the matrix $\boldsymbol{A}$ in decreasing order;
2: Construct the perturbed eigenvector matrix $\widetilde{\boldsymbol{V}}$:
    (i) Generate a $p \times r$ random matrix $\mathbf{B}$ and calculate the perturbed matrix $\boldsymbol{V}^* = \boldsymbol{V} + \mathbf{B}$, where $\mathbf{B} = (\mathbf{b}_1, \cdots, \mathbf{b}_r) \in \mathbb{R}^{p \times r}$, each entry of the vector $\mathbf{b}_i$ is from $Lap(2\sqrt{p}/\epsilon_{1i})$, $\epsilon_{1i} = \omega_i \epsilon_1$, $\omega_i > 0$ and $\sum_{i=1}^r \omega_i = 1$;
    (ii) Calculate the orthogonal matrix $\widetilde{\boldsymbol{V}} \in \mathbb{R}^{p \times r}$ of the matrix $\boldsymbol{V}^*$ by singular value decomposition;
3: Calculate the estimators of the factor matrix and factor loading matrix:
    (I) Generate a $n \times r$ random matrix $\boldsymbol{C}$, where each entry of $\boldsymbol{C}$ is sampled from a $Lap(2r/\epsilon_2)$;
    (II) Calculate factor loading matrix $\widetilde{\boldsymbol{\Lambda}} = \widetilde{\boldsymbol{V}}$ and factor matrix $\widetilde{\mathbf{W}} = \boldsymbol{X}\widetilde{\boldsymbol{V}} + \boldsymbol{C}$;
4: Return: Synthesized data $\widetilde{\boldsymbol{X}} = \widetilde{\mathbf{W}}\widetilde{\boldsymbol{\Lambda}}^\top$.
```

Note that in Step 2 (i), we can consider different weights ω_i for the allocation of the privacy budget ϵ_1 for example, $\omega_i = 1/r$ or $\omega_i = \nu_i / \sum_{j=1}^r \nu_j$ incorporating the magnitudes of the eigenvalues, where ν_j is the j-th largest eigenvalue of the matrix $\boldsymbol{X}^\top \boldsymbol{X}$. Based on the tail bound for an ensemble matrix in Tao (2012), we have Frobenius norms of the noise matrices $\mathbf{B}$ in Step 2 (i) and $\boldsymbol{C}$ in Step 3 (I) for the given r satisfying $\|\mathbf{B}\|_F = O(p/\epsilon)$ and $\|\boldsymbol{C}\|_F = O(\sqrt{n}/\epsilon)$ with a high probability, which are important for the utility guarantees.

3.3 Implementation for Mixed-type Data

In the following, we consider mixed-type data containing continuous, ordinal categorical, and nominal categorical data and try to keep their original data types in the released data. To this end, we import data type conversion into the proposed differential privacy method for data-type consistency.

We consider an original dataset $\boldsymbol{X} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n)^\top$ with mixed-type data including continuous, ordinal and nominal categorical variables. Denote the ith sample as

$$\boldsymbol{x}_i = (y_{i,1}, \ldots, y_{i,p_1}, z_{i,p_1+1}, \ldots, z_{i,p_1+p_2}, u_{i,p_1+p_2+1}, \ldots, u_{i,p_1+p_2+p_3})^\top$$

for $i = 1, \ldots, n$, where $y_{i,j}, z_{i,j}$ and $u_{i,j}$ denote continuous, ordinal and nominal categorical variables, respectively. The classical factor model (1) mostly focuses on continuous variables (Bai and Ng, 2002; Bai, 2003), which brings limitations to handle mixed-type data. As the observed data contain discrete data, the generalized factor model (Skrondal and Rabe-Hesketh, 2004; Goldstein and Browne, 2005) or link functions between continuous and discrete variables (Song et al., 2013) are considered. To provide a unified factor model framework for different types of data, we adopt link functions between continuous and discrete variables and transform mixed-type data to continuous data before implementing the above proposed algorithm.

Based on the framework of Song et al. (2013), we construct an underlying vector

$$\boldsymbol{x}_i^* = (y_{i,1}^*, \ldots, y_{i,p_1}^*, z_{i,p_1+1}^*, \ldots, z_{i,p_1+p_2}^*, \mathbf{u}_{i,p_1+p_2+1}^{*\top}, \ldots, \mathbf{u}_{i,p_1+p_2+p_3}^{*\top})^\top,$$

which is linked to the original vector data $\boldsymbol{x}_i$ as follows:

$$\begin{cases} y_{ij} &= h_{1j}(y_{ij}^*), \quad j = 1, \ldots, p_1, \\ z_{ij} &= h_{2j}(z_{ij}^*), \quad j = p_1 + 1, \ldots, p_1 + p_2, \\ u_{ij} &= h_{3j}(\mathbf{u}_{ij}^*), \quad j = p_1 + p_2 + 1, \ldots, p_1 + p_2 + p_3, \end{cases} \tag{5}$$

where h_{1j}, h_{2j} and h_{3j}'s correspond to identity, threshold and multinomial probit link functions, respectively. The identity link function $h_{1j}(\cdot)$ keeps the continuous variables y_{ij} invariant, that is, $y_{ij} = h_{1j}(y_{ij}^*) = y_{ij}^*$.

For nominal categorical variables u_{ij} with M_j categories, we assume that u_{ij} takes values from $\{0, 1, \ldots, M_j - 1\}$. The u_{ij} is transformed to a continuous vector $\mathbf{u}_{ij}^* = (u_{ij,1}^*, \ldots, u_{ij,M_j-1}^*)^\top \in \mathbb{R}^{M_j-1}$ via the following multinomial probit link function

$$u_{ij} = h_{3j}(\mathbf{u}_{ij}^*) = \begin{cases} 0, & \text{if } \max(\mathbf{u}_{ij}^*) \le 0, \\ l, & \text{if } \max(\mathbf{u}_{ij}^*) = u_{ij,l}^* > 0, \end{cases} \tag{6}$$

where each element of $\mathbf{u}_{ij}^*$ is sampled from a truncated standard normal distribution with the truncation range $(-\infty, 0)$ if $u_{ij} = 0$, and each element of $\mathbf{u}_{ij}^*$ is sampled from a standard normal distribution so that $\max(\mathbf{u}_{ij}^*) = u_{ijl}^* > 0$ if $u_{ij} = l$. Based on the continuous vector $\boldsymbol{u}_{ij}^*$ for $i = 1, \ldots, n; j = p_1 + p_2 + 1, \ldots, p_1 + p_2 + p_3$, the proposed differentially private algorithm can generate continuous synthetic data, denoted as $\widetilde{\boldsymbol{u}}_{ij}^*$. Based on the link function (6), we can obtain the nominal synthetic data $\widetilde{u}_{ij}$ from the index and sign of the largest component of $\widetilde{\boldsymbol{u}}_{ij}^*$. Since the transformation of the continuous vectors $\widetilde{\boldsymbol{u}}_{i,j}^*$ to nominal values $\widetilde{u}_{ij}$ only depends on the index and the sign of the largest component of the synthetic data $\widetilde{\boldsymbol{u}}_{ij}^*$ with privacy protection, the transformation does not impose the risk of leakage.

For ordinal variables z_{ij} with integer values in $\{0, 1, \ldots, L_j - 1\}$, a threshold link function $h_{2j}(\cdot)$ is defined as the following:

$$z_{ij} = h_{2j}(z_{ij}^*) = \sum_{l=0}^{L_j-1} l \cdot I(\tau_{j,l} \le z_{ij}^* < \tau_{j,l+1}), \tag{7}$$

where $I(\cdot)$ is an indicator function, and takes a value of 1 if $\tau_{j,l} \leq z^*_{ij} < \tau_{j,l+1}$ and 0 otherwise. Here, $\{-\infty = \tau_{j,0} < \tau_{j,1} < \cdots < \tau_{j,L_j} = +\infty\}$ is a set of thresholds defining the L_j categories. The set of thresholds can be estimated by converting the cumulative proportions of the observed data z_{ij} (Song et al., 2013). That is, $\tau_{j,l} = \Phi^{-1}(f_{j,l})$ for $l = 1, 2, \ldots, L_j - 1$, where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal $N(0, 1)$, and $f_{j,l}$ is the cumulative frequency of categories with $z_{ij} < l$. With the set of estimated thresholds, if $z_{ij} = l$, z^*_{ij} is generated from a truncated standard normal distribution with the truncation range $(\tau_{j,l}, \tau_{j,l+1})$ as a continuous transformation of z_{ij}.

With the continuous data z^*_{ij} for $i = 1, \ldots, n; j = p_1 + 1, \ldots, p_1 + p_2$, the proposed differentially private algorithm can generate continuous synthetic data, denoted as $\widetilde{z}^*_{ij}$. Since the set of thresholds in the link function (7) are constructed based on the cumulative frequency of the original variables z_{ij}, we need to build differentially private cumulative frequencies for the privacy-protecting transformation from the continuous synthetic value $\widetilde{z}^*_{ij}$ to ordinal value via the link function (7). There are many methods for frequency or cumulative frequency estimation with privacy protection, for example Wang et al. (2017), Cormode et al. (2021), and Arcolezi (2022). However, these methods are locally differentially private for frequency estimation. Alternatively, we build a differential privacy approach for cumulative frequency based on the Laplace mechanism. First, we establish the sensitivity of the cumulative frequency as follow.

Lemma 7 *Denote $\boldsymbol{f}_j : \mathbb{R}^n \to \mathbb{R}^{L_j+1}$ for $j = p_1+1, p_1+2, \ldots, p_1+p_2$ as the cumulative frequencies of ordinal categories, where $\boldsymbol{f}_j(\boldsymbol{z}_j) = (f_{j0}(\boldsymbol{z}_j), f_{j1}(\boldsymbol{z}_j), \cdots, f_{j,L_j-1}(\boldsymbol{z}_j), f_{j,L_j}(\boldsymbol{z}_j))$, $f_{jl}(\boldsymbol{z}_j) = \sum_{i=1}^n I(z_{ij} < l)/n$ and $\boldsymbol{z}_j \in \mathbb{R}^n$. Then the l_1-sensitivity of the cumulative frequency function $\boldsymbol{f}_j$ satisfies*

$$\Delta \boldsymbol{f}_j = \max_{\boldsymbol{z}_j : |\boldsymbol{z}_j - \boldsymbol{z}'_j|_1 = 1} \|\boldsymbol{f}_j(\boldsymbol{z}_j) - \boldsymbol{f}_j(\boldsymbol{z}'_j)\|_1 \leq (L_j - 1)/n,$$

where $|\boldsymbol{z}_j - \boldsymbol{z}'_j|_1 = 1$ denotes that $\boldsymbol{z}_j$ and $\boldsymbol{z}'_j$ contain at most one different record.

Lemma 7 provides the sensitivity of the cumulative frequency function. Based on the Laplace mechanism and the above lemma, we construct the cumulative frequency with privacy protection for each ordinal variable via the following Algorithm 2, which is proved to be differentially private. See Lemma 13.

Algorithm 2 Differentially private cumulative frequency functions releasing algorithm

```
Input:
    Original data $\mathbf{z}_j \in \mathbb{R}^n$ and privacy budget $\epsilon$;
Output:
    Perturbed cumulative frequency $\widetilde{\boldsymbol{f}}_j(\mathbf{z}_j)$;
Calculate cumlative frequencies $\boldsymbol{f}_j(\boldsymbol{z}_j) = (f_{j0}(\boldsymbol{z}_j), f_{j1}(\boldsymbol{z}_j), \cdots, f_{j,L_j-1}(\boldsymbol{z}_j), f_{j,L_j}(\boldsymbol{z}_j))$, where $f_{jl}(\boldsymbol{z}_j) = \sum_{i=1}^{n} I(z_{ij} < l)/n$;
Construct the perturbed cumulative frequency $\widetilde{\boldsymbol{f}}_j$ with privacy protection:
    (i) Construct a noise vector $\boldsymbol{e} = (e_0, e_1, \cdots, e_{L_j})$ with ascending entries from $Lap(\Delta \boldsymbol{f}_j/\epsilon)$, where $e_0 \leq e_1 \leq e_2 \leq \cdots \leq e_{L_j}$;
    (ii) Construct the perturbed cumulative frequency
        $$\widetilde{\boldsymbol{f}}_j(\boldsymbol{z}_j) = \frac{(\boldsymbol{f}_j(\boldsymbol{z}_j) + \boldsymbol{e}) - \min\{\boldsymbol{f}_j(\boldsymbol{z}_j) + \boldsymbol{e}\}}{\max\{\boldsymbol{f}_j(\boldsymbol{z}_j) + \boldsymbol{e}\} - \min\{\boldsymbol{f}_j(\boldsymbol{z}_j) + \boldsymbol{e}\}};$$
Return: Perturbed cumulative frequency $\widetilde{\boldsymbol{f}}_j(\mathbf{z}_j)$.
```

Note that the perturbed cumulative frequency $\widetilde{\boldsymbol{f}}_j(\mathbf{z}_j)$ still satisfies $0 = f_{j0} \leq f_{j1} \leq \cdots \leq f_{j,L_j-1} \leq f_{j,L_j} = 1$. We utilize the cumulative frequencies $\tilde{f}_{j,l}$ with privacy protection to obtain a set of thresholds, that is, $\tilde{\tau}_{j,l} = \Phi^{-1}(\tilde{f}_{j,l})$. Based on the post-processing property, the set of thresholds is ϵ-differentially private, and the transformation based on the following link function (8) is also privacy-protected.

$$\tilde{z}_{ij} = \tilde{h}_{2j}(z_{ij}^*) = \sum_{l=0}^{L_j-1} l \cdot I(\tilde{\tau}_{j,l} \leq z_{ij}^* < \tilde{\tau}_{j,l+1}). \tag{8}$$

Combining with the link functions (5)-(8), we build the following differentially private data-releasing algorithm for mixed-type data, which can obtain final release data with the same data types as the original data.

Algorithm 3 Differentially private data-releasing algorithm for mixed-type data

```
Input:
    Original data $\boldsymbol{X}$, privacy budget $\epsilon = \epsilon_1 + \epsilon_2 + \epsilon_3$ and the number of factors $r$;
Output:
    Synthesized data $\widetilde{\boldsymbol{X}}$;
Construct link functions (5) based on $\boldsymbol{X}$ and generate continuous data $\boldsymbol{X}^*$ via link functions (5);
Execute Algorithm 1 on the data $\boldsymbol{X}^*$ with privacy budget $\epsilon_1$ and $\epsilon_2$, and obtain continuous synthetic data $\widetilde{\boldsymbol{X}}^*$;
Execute Algorithm 2 with privacy budget $\epsilon_3$, and construct the threshold link function (8);
Transform the data $\widetilde{\boldsymbol{X}}^*$ to mixed-type synthetic data $\widetilde{\boldsymbol{X}}$ via the link functions (6)-(8);
Return: Synthesized data $\widetilde{\boldsymbol{X}}$.
```

Note that Algorithm 3 involves privacy budget allocation, that is, ϵ_1, ϵ_2 and ϵ_3. Based on the composition property, Algorithm 3 is always ϵ-differentially private for any privacy budget allocation. Moreover, the utility of the synthetic data obtained from Algorithm 3 is also relatively robust for the budget allocation. See the numerical studies in Section 4.

3.4 Privacy Guarantee and Utility Guarantee

In this subsection, we provide the theoretical properties of Algorithm 1-Algorithm 3, which consist of the privacy guarantees and the utility guarantee. The privacy guarantee is the basic requirement of a differential privacy algorithm, and the utility guarantee indicates how effective the algorithm is against the non-private version. We firstly present two lemmas for the privacy guarantee of the proposed algorithms.

Lemma 8 *Denote $\boldsymbol{\mu}_i : \mathbb{R}^{n\times p} \to \mathbb{R}^p$ for $i = 1, 2, \ldots, r$ as the i-th eigenvector corresponding to the i-th largest eigenvalue of the matrix $\boldsymbol{X}^\top \boldsymbol{X}$ in decreasing order, where $\boldsymbol{X} \in \mathbb{R}^{n\times p}$. Then $\boldsymbol{\mu}_i(\boldsymbol{X}) + \mathbf{b}$ is ϵ_{1i}-differentially private, where $\mathbf{b} \in \mathbb{R}^p$ is from a Laplace distribution $Lap(2\sqrt{p}/\epsilon_{1i})$.*

Lemma 8 shows that the perturbed eigenvector is ϵ_{1i}-differentially private and has the added noise level $O(p/\epsilon_{1i})$ for $i = 1, \cdots, r$. Based on the composition property, we can prove that the perturbed matrix $\boldsymbol{V}^*$ from Step 2 (i) is differentially private in the following lemma.

Lemma 9 *Denote $\boldsymbol{V} : \mathbb{R}^{n\times p} \to \mathbb{R}^{p\times r}$ as the eigenvectors matrix corresponding to the first r largest eigenvalues of the matrix $\boldsymbol{X}^\top \boldsymbol{X}$ in decreasing order, where $\boldsymbol{X} \in \mathbb{R}^{n\times p}$. Then the algorithm $\mathcal{M}(\boldsymbol{X}) = \boldsymbol{V}(\boldsymbol{X}) + \mathbf{B}$ is ϵ_1-differentially private, where $\mathbf{B} = (\mathbf{b}_1, \cdots, \mathbf{b}_r) \in \mathbb{R}^{p\times r}$, each entry of $\mathbf{b}_i$ is from a Laplace distribution $Lap(2\sqrt{p}/\epsilon_{1i})$ $(i = 1, \cdots, r)$, $\epsilon_{1i} = \omega_i \epsilon_1$, $\omega_i > 0$ and $\sum_{i=1}^r \omega_i = 1$.*

Lemma 9 implies that the perturbed eigenvector matrix $\boldsymbol{V}^*$ from Step 2 (i) in the Algorithm 1 satisfies ϵ_1-differential privacy. Since $\mathbf{B} \in \mathbb{R}^{p\times r}$ and each entry of $\mathbf{b}_i$ is from $Lap(2\sqrt{p}/\epsilon_{1i})$ $(i = 1, \cdots, r)$, the level of the noise added to $\boldsymbol{V}^*$ is $O(p\sqrt{\sum_{i=1}^r 1/\omega_i^2}/\epsilon_1)$. Since $\omega_i > 0$ and $\sum_{i=1}^r \omega_i = 1$, the following inequality holds: $\sqrt{\sum_{i=1}^r 1/\omega_i^2} \geq r\sqrt{r}/\sum_{i=1}^r \omega_j = r\sqrt{r}$. The equality holds when $\omega_i = 1/r$, and the added noise level reaches the lower bound $O(pr\sqrt{r}/\epsilon_1)$. If we take the eigenvalues into account and set $\omega_i = \nu_i/\sum_{j=1}^r \nu_j$, then the level of the noise is $O(p(\sum_{j=1}^r \nu_j)\sqrt{\sum_{j=1}^r 1/\nu_j^2}/\epsilon_1)$. Based on Lemma 8 and Lemma 9, we can establish the privacy guarantee of the Algorithm 1 as follow.

Theorem 10 ***(Privacy guarantee for continuous data)*** *Algorithm 1 above returns a privacy-preserving matrix $\widetilde{\boldsymbol{X}}$ such that $\widetilde{\boldsymbol{X}}$ satisfies $(\epsilon_1 + \epsilon_2)$-differential privacy.*

Theorem 10 guarantees that Algorithm 1 satisfies differential privacy given a privacy budget ϵ, where ϵ is split into ϵ_1 and ϵ_2 for the privacy budget of the factor loading matrix and the privacy budget of the factor matrix, respectively. It implies that the proposed method can protect the original data from being identified regardless of whether individual data contain a particular sample or not. Consequently, the synthetic data have a low risk of leakage. In the following, we derive an upper bound of the proposed method for the utility guarantee. Firstly, we present a lemma for the utility guarantee.

Lemma 11 *The matrix $\widetilde{\boldsymbol{V}}$ from Algorithm 1 satisfies with a high probability*

$$\|\widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{V}}^\top - \boldsymbol{V}\boldsymbol{V}^\top\|_F \le O(\sqrt{r}p/\epsilon_1).$$

Lemma 11 establishes the top-r subspace closeness of the matrix $\widetilde{\boldsymbol{V}}$ obtained from Algorithm 1, which implies that the matrix $\widetilde{\boldsymbol{V}}$ not only captures a large amount of variance, but is also close to the matrix spanned by the top r eigenvectors subspace $\boldsymbol{V}$ of $\boldsymbol{X}^\top\boldsymbol{X}$. Dwork et al. (2014), Jiang et al. (2016), and Imtiaz and Sarwate (2018) also provided similar upper bounds of the utility for the perturbed subspace, but they require that the matrix $\boldsymbol{X}^\top\boldsymbol{X}$ have a large spectrum gap between the r-th singular value and the $(r+1)$-th singular value, as otherwise the top r subspace space is not uniquely defined. In comparison, Lemma 11 only requires that the first r eigenvalues satisfy condition (3) in Section 3.1. Based on Lemma 11, we can derive an upper bound of the utility for Algorithm 1.

Theorem 12 ***(Upper bound on utility for continuous data)*** *Under the fixed number of factors r, the matrix $\widetilde{\boldsymbol{X}}$ from Algorithm 1 satisfies with a high probability*

$$\|\widetilde{\boldsymbol{X}} - \boldsymbol{X}\|_F \le \|\widehat{\boldsymbol{X}} - \boldsymbol{X}\|_F + O\left(\frac{p+\sqrt{n}}{\epsilon}\right) \le O\left(\sqrt{n} + \frac{p+\sqrt{n}}{\epsilon}\right). \tag{9}$$

Theorem 12 indicates that the utility bound of the releasing data matrix in regard to the original data depends on sample size, dimension, the number of factors and privacy budgets. Note that if $\|\widetilde{\boldsymbol{X}} - \boldsymbol{X}\|_F \le \|\widehat{\boldsymbol{X}} - \boldsymbol{X}\|_F + \phi$, we refer to the parameter ϕ as the additive error of the releasing matrix $\widetilde{\boldsymbol{X}}$. If the dimension of variables satisfies $p = O(n^\kappa)$ and $\kappa \in (0, 1/2)$, then the $\sqrt{n}/\epsilon$ would dominate the p/ϵ so that the additive error is $O(\sqrt{n}/\epsilon)$ which is lower than that of existing differentially private low-rank approximations, e.g., Arora et al. (2018) or Upadhyay (2018). In addition, Arora et al. (2018) provide the additive error of the (ϵ, δ)-differentially private low-rank approximation based on the operator norm of the matrix. Based on the inequality relationship of the Frobenius norm and the operator norm, their corresponding additive error based on the Frobenius norm is $O(p^{1/2}(n+p)\text{polylog}(n)\log^2(1/\delta)/\epsilon)$ for the fixed r. Since the definitions of the neighboring matrix for Arora et al. (2018) and our approach are rather different, we cannot compare to their bound directly. If we ignore the definition difference in comparison, it is clear that their additive error is larger than that of our proposed method.

Moreover, Upadhyay (2018) provide the additive error of their low rank approximation of the streamed matrix with (ϵ, δ)-differential privacy. That is, $O(\sigma_{\min}\sqrt{n} + \sqrt{p\ln(1/\delta)}/\epsilon)$, where $\sigma_{\min} = 16\log(1/\delta)\sqrt{t(1+\phi)(1-\phi)^{-1}\ln(1/\delta)}/\epsilon$, $t = O(\max(r, \alpha^{-1})\alpha^{-1}\log(1/\delta))$, and $\phi \in (0, 1)$. Similarly, their definition of the neighboring matrix is different from our proposed method. Moreover, their method deals with the streamed matrix data. If we ignore the definition difference and streamed matrix data type, their additive error is also larger than that of the proposed method.

In the following, we derive the privacy guarantees of Algorithm 2-3.

Lemma 13 *Algorithm 2 returns a privacy-preserving perturbed cumulative frequency* $\widetilde{f}_j(\mathbf{z}_j)$ *such that* $\widetilde{f}_j(\mathbf{z}_j)$ *satisfies* ϵ*-differential privacy.*

Lemma 13 shows that Algorithm 2 is ϵ-differential privacy, which can be proved based on the framework of the mLaplace mechanism. The following theorem shows the privacy guarantee of Algorithm 3.

Theorem 14 ***(Privacy guarantee for mixed-type data)*** *Algorithm 3 returns a privacy-preserving matrix* $\widetilde{\boldsymbol{X}}$ *in that* $\widetilde{\boldsymbol{X}}$ *satisfies* $(\epsilon_1 + \epsilon_2 + \epsilon_3)$*-differential privacy.*

Theorem 14 can be obtained based on privacy guarantees of Algorithms 1 and 2 and the composition property of differential privacy. Next, we present the utility guarantee of Algorithm 3 as follows.

Theorem 15 ***(Upper bound on utility for mixed-type data)*** *Under the assumptions that the number of factors* r *is fixed,* $\max_{1\le j\le p_2}(L_j - 1)^2 = O(1)$ *and* $\max_{1\le j\le p_3}(M_j - 1)^2 = O(1)$*, the matrix* $\widetilde{\boldsymbol{X}}$ *obtained from Algorithm 3 satisfies with a high probability*

$$\|\widetilde{\boldsymbol{X}} - \boldsymbol{X}\|_F \le O\left(\sqrt{n} + \frac{p_1 + \sqrt{n}}{\epsilon} + \sqrt{np_2} + \sqrt{np_3}\right).$$

Theorem 15 indicates that the utility bound of the releasing mixed-type data in regard to the original data depends on sample size, dimension, the number of factors, privacy budgets, and the number of classes for ordinal variables. If $p_1 = O(n^{\kappa})$, $p_2 = O(n^{\alpha})$, $p_3 = O(n^{\ell})$ $(\kappa, \alpha, \ell \in (0, 1/2))$, the right hand side of the above inequality is $O(n^{(1+\max(\alpha,\ell)/2)} + n^{1/2}/\epsilon)$. Similar to the discussion of Theorem 12, the error bound of the proposed method for mixed-type data is lower than that of existing low-rank approximation approaches with differential privacy.

4. Simulation Studies

In this section, we use simulated datasets to evaluate the performance of the proposed method (DPFM) compared with the data releasing methods based on the PCA, including the differential PCA-based privacy preserving data publishing (PCAPPD, Jiang et al., 2013) and differentially private data release via random projections (DPRP, Gondara and Wang, 2020). Moreover, we also consider the comparison methods from NIST PSCR, including synthetic data algorithms based on Bayesian network (PrivBay, Ping et al., 2017) and graphical model (PrivPGM, Mckenna et al., 2019; McKenna et al., 2021). The PCAPPD method uses one-hot encoding to perform the transformation between discrete and continuous variables. The DPRP and PrivBay methods treat all variables as continuous variables. Since the PrivPGM only deals with discrete data, the continuous variables are discretized into 100 bins. As a reference, we consider the performance of the original data (Original) as a baseline.

We consider classification problems to assess the utilities of the dataset released. We use three classification methods for evaluation including support vector machine (SVM), random forests (RF) and K-nearest neighbors algorithm (KNN). For evaluating the utility of the synthetic data generated via considered differentially private methods, we calculate the accuracy, F1 score and the area under the precision recall curves (AUC) of each classification method and calculate the averages for three classifications over different criteria.

To empirically evaluate the quality of the generated dataset, we introduce a synthetic training set, synthetic testing set and raw validating set. Specifically, we use all raw data to construct synthetic data and use 80% of the synthetic data into training data and the remaining 20% into testing data. We use the partial original data corresponding to the testing data as validation data. If the prediction performances on both the original validating set and the synthetic testing set are high for models trained on the synthetic training set, we can infer that the synthetic data capture the original data information well.

We consider a dataset $\boldsymbol{X} = (\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3, \boldsymbol{X}_4, \boldsymbol{X}_5) = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n)^\top \in \mathbb{R}^{n\times p}$, where continuous data $\boldsymbol{X}_1 \in \mathbb{R}^{n\times p_1}$ are generated from a multivariate normal distribution with mean 0 and covariance matrix $\boldsymbol{\Sigma}$, ordinal data $\boldsymbol{X}_2 \in \mathbb{R}^{n\times p_2}$ randomly from $\{0, 1, 2\}$, ordinal data $\boldsymbol{X}_3 \in \mathbb{R}^{n\times p_3}$ randomly from $\{0, 1\}$, nominal data $\boldsymbol{X}_4 \in \mathbb{R}^{n\times p_4}$ randomly from $\{0, 1, 2, 3\}$ and nominal data $\boldsymbol{X}_5 \in \mathbb{R}^{n\times p_5}$ randomly from $\{0, 1\}$. For generating labels $\mathbf{y} = (y_1, y_2, \cdots, y_n)^\top$, we transform ordinal and nominal variables into dummy variables, and generate the label y_i from a Bernoulli distribution with a probability $\Pr(y_i = 1) = \exp(\beta_0 + \boldsymbol{\beta}^\top \mathbf{z}_i)/(1 + \exp(\beta_0 + \boldsymbol{\beta}^\top \mathbf{z}_i))$, where $\mathbf{z}_i$ is the ith sample record with dummy variables after transformation, $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^\top, \cdots, \boldsymbol{\beta}_5^\top)^\top$, $\boldsymbol{\beta}_1 \in \mathbb{R}^{p_1}$, $\boldsymbol{\beta}_2 \in \mathbb{R}^{2p_2}$, $\boldsymbol{\beta}_3 \in \mathbb{R}^{p_3}$, $\boldsymbol{\beta}_4 \in \mathbb{R}^{3p_4}$ and $\boldsymbol{\beta}_5 \in \mathbb{R}^{p_5}$.

We consider a covariance matrix $\boldsymbol{\Sigma}$ with a common marginal variance 1 and correlation structures: AR-1 with correlation $\rho = 0.95$ and exchangeable (Exch) with correlation $\rho = 0.95^2$. We set sample size $n = 200$ and consider the following two settings of dimension:

(I) $p = 22$, where $p_1 = 6$, $p_2 = 4$, $p_3 = 4$, $p_4 = 4$ and $p_5 = 4$. Correspondingly, $\beta_0 = 1.5$, $\boldsymbol{\beta}_1 = (3, 2.5, 2, 1.5, 1, 0.5)^\top$, $\boldsymbol{\beta}_2 = (0.3, 1, 0.1, 1, 0.2, 2, 0.5, 3)^\top$, $\boldsymbol{\beta}_3 = (1.5, 0.8, 0.5, 1.8)^\top$, $\boldsymbol{\beta}_4 = (0.5, 1, 3, 0.5, 1, 2, 0.5, 1, 0.3, 0.5, 1, 0.2)^\top$, and $\boldsymbol{\beta}_5 = (1, 1.5, 0.5, 0.5)^\top$.

(II) $p = 80$, where $p_1 = 20$, $p_2 = 15$, $p_3 = 15$, $p_4 = 15$, and $p_5 = 15$. Correspondingly, $\beta_0 = 1.5$, $\boldsymbol{\beta}_1 = (3, 2.5, 2, 1.5, 1, 0.5, 0, \cdots, 0)^\top$, $\boldsymbol{\beta}_2 = (0.3, 1, 0.1, 1, 0.2, 2, 0.5, 3, 0, \cdots, 0)^\top$, $\boldsymbol{\beta}_3 = (1.5, 0.8, 0.5, 1.8, 0, \cdots, 0)^\top$, $\boldsymbol{\beta}_4 = (0.5, 1, 3, 0.5, 1, 2, 0.5, 1, 0.3, 0.5, 1, 0.2, 0, \cdots, 0)^\top$, and $\boldsymbol{\beta}_5 = (1, 1.5, 0.5, 0.5, 0, \cdots, 0)^\top$.

For generating synthetic data satisfying various privacy protection requirements, we consider different privacy budgets $\epsilon \in \{0.1, 0.5, 1, 1.5, \cdots, 5\}$. For the proposed method, we consider different allocations of the privacy budgets $(\epsilon_1, \epsilon_2, \epsilon_3) = (\frac{\epsilon}{3}, \frac{\epsilon}{3}, \frac{\epsilon}{3}), (\frac{2\epsilon}{3}, \frac{\epsilon}{6}, \frac{\epsilon}{6}), (\frac{\epsilon}{6}, \frac{2\epsilon}{3}, \frac{\epsilon}{6})$ and $(\frac{\epsilon}{6}, \frac{\epsilon}{6}, \frac{2\epsilon}{3})$, and denote the corresponding methods as DPFM_1, DPFM_2, DPFM_3 and DPFM_4, respectively. For the DPRP method, we fix a small $\delta = n^{-5}$ to be comparable. We set the threshold $c = 0.8$ to select the number of factors. A decreasing privacy budget indicates an increasing privacy protection requirement. The same experiments are replicated 100 times. Figures 1 - 4 show the results for various budgets. Tables 3 and 4 show the results as $\epsilon = 0.1$ and 5 in the supplementary materials.

From Figures 1 - 4, we can observe that the proposed methods under varying budget allocations outperform all comparison methods for all settings. The proposed methods achieve higher accuracy, F1 scores and AUC than other comparing methods for various privacy budgets. We observe that DPFM_1, DPFM_2, DPFM_3 and DPFM_4 have similar performance for all criteria in setting (I) and (II). These numerical findings imply that the proposed method is robust to the privacy budget division. Moreover, we can observe that the performances of the proposed methods based on three criteria are also robust to the privacy budget. In contrast, the performances of the four comparison methods based on three criteria fluctuate around low values. This indicates that the proposed methods, regardless of the privacy budget divisions, can generate synthetic data with much higher utility for classification problems for correlated datasets with certain privacy protection requirements.

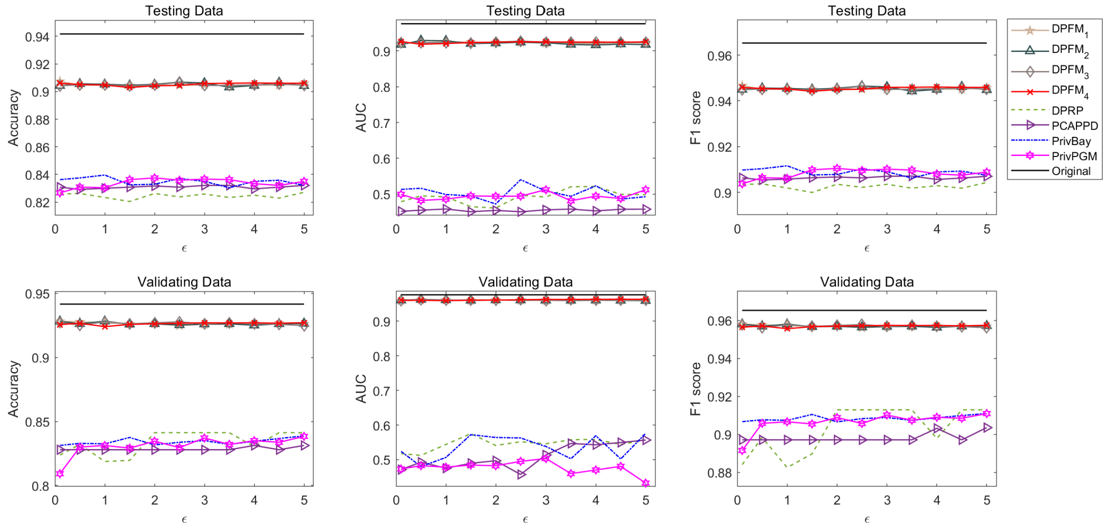

Figure 1: Results of three classifiers on testing synthetic data from eight algorithms and validating data in setting (I) with AR-1 correlation matrix.

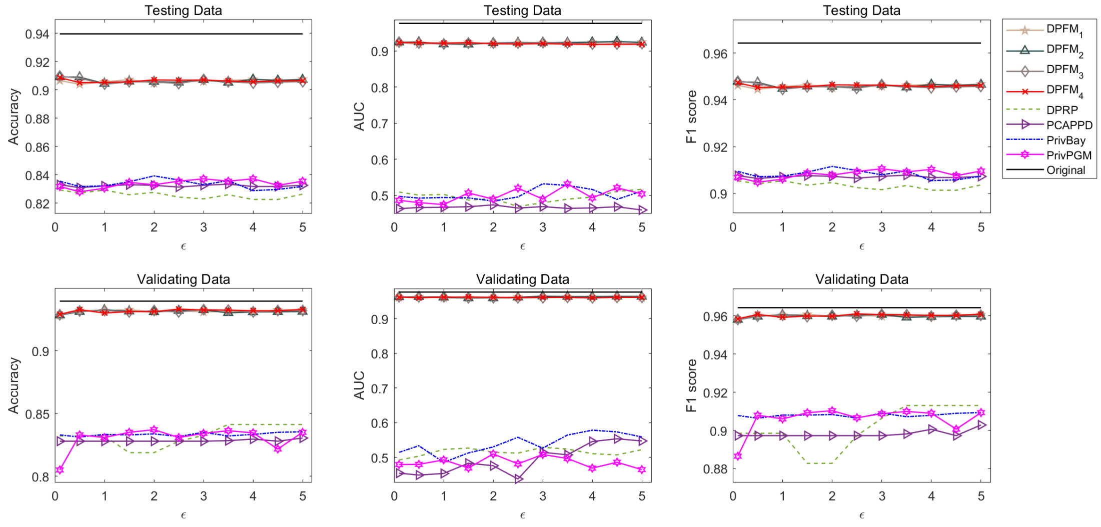

Figure 2: Results of three classifiers on testing synthetic data from eight algorithms and validating data in setting (I) with exchangeable correlation matrix.

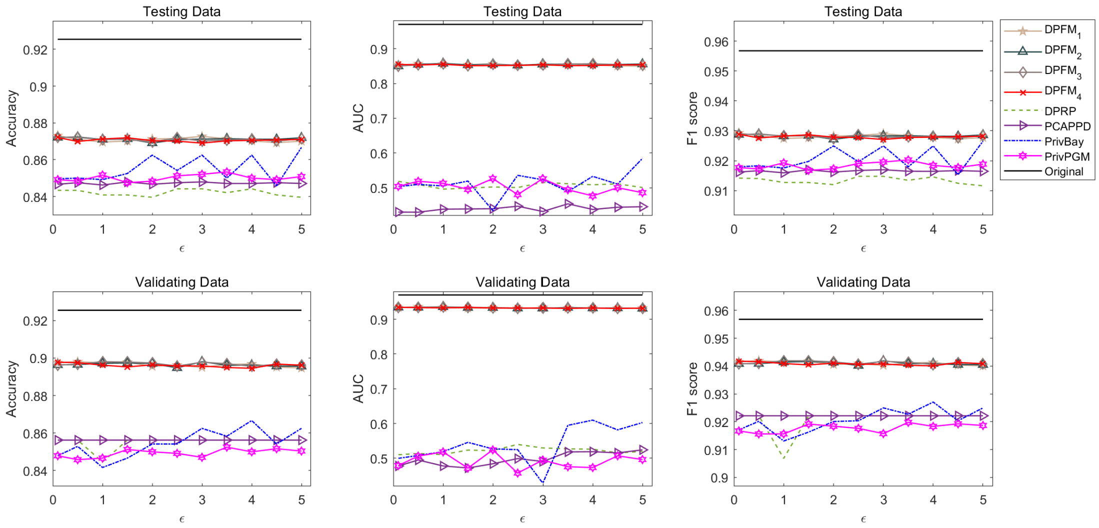

Figure 3: Results of three classifiers on testing synthetic data from eight algorithms and validating data in setting (II) with AR-1 correlation matrix.

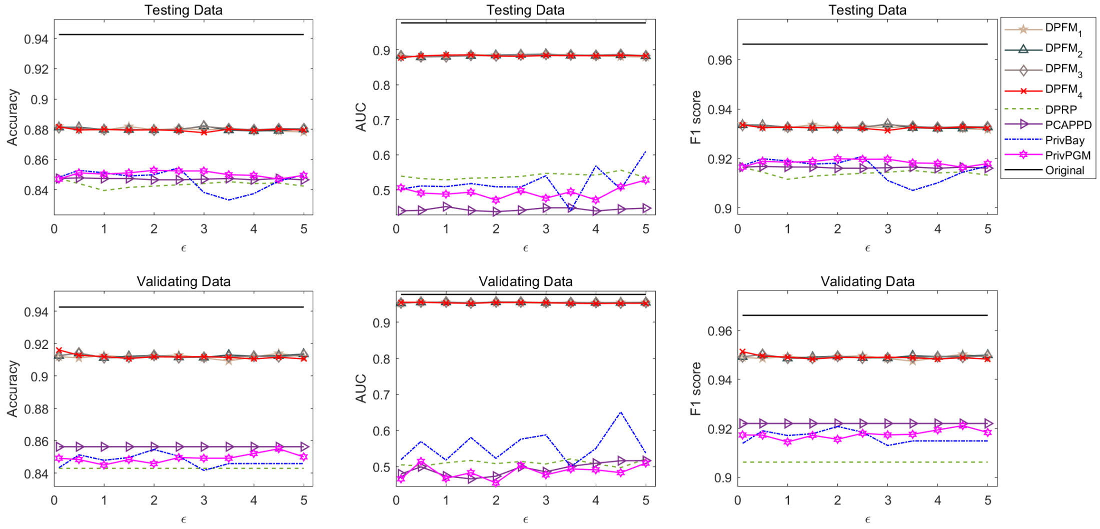

Figure 4: Results of three classifiers on testing synthetic data from eight algorithms and validating data in setting (II) with exchangeable correlation matrix.

5. Real Data Examples

In this section, we evaluate the performance of the proposed method for real data examples through comparison of the PCAPPDP, DPRP, PrivBay and PrivPGM methods similar to Section 4. We use three datasets from the UCI machine learning repository to evaluate the utility of the synthetic data for classification tasks, including census income data, absenteeism at work data and breast cancer data. As a reference, we also consider the performance of the original data (Original) as a baseline. Similar to Section 4, we split the synthetic training set, synthetic testing set and raw validating set, and consider three classification methods including SVM, RF and KNN methods. We calculate the accuracy, F1 score and AUC value to evaluate the utility of the synthetic data based on several differentially private methods.

Census Income Data: The dataset from the U.S. Census Bureau has been generally used to predict whether a given adult makes more than $\$50,000$ a year. In the dataset, there are 48842 instances containing 23.93% of incomes labeled with more than $\$50,000$ and 76.07% of incomes below $\$50,000$. There are 15 features of each instance, including age, weight, marital status, native country, race, sex, label of income, and so on. After removing the subjects with missing data, there are 33916 subjects. We use 11 demographic variables to predict the label of income: whether a person earned more than $\$50,000$ per year. The details of the variables are shown in Table 1.

Table 1: Variables from the Census Income Data.

Variable Name	Data Type	Range
age	continuous	(17,90)
education-num	continuous	(1,16)
hour-per-week	continuous	(1,99)
capital loss	ordinal	0,1
capital gain	ordinal	0,1
label of income	nominal	0,1
marital status	nominal	Couple, Single
native country	nominal	US, Non-US
gender	nominal	Male, Female
work class	nominal	Govt, Private, Self-employed, Without pay
race	nominal	White, Amer-Indian-Eskimo, Asian-Pac-Islander, Black, Other
relationship	nominal	Unmarried, Wife, Husband, Not-in-family, Own-child, Other-relative

Absenteeism at Work Data: The dataset is created with records of absenteeism at work from July 2007 to July 2010 at a courier company in Brazil. In this dataset, there are 740 instances containing 46% of individuals with absenteeism time more than four hours per month and 54% of individuals with absenteeism time below four hours per month. There are 21 features of each instance, including individual identification, number of children (0,1,2,larger than 2), number of pets (0,1,larger than 1), education (high school, graduate, master or above), day of the week (Monday, Tuesday, Wednesday, Thursday, Friday), and so on. We use 19 variables to predict the label of absenteeism time: whether a person is absent more than four hours per month. The details of the variables are shown in Table 2.

Table 2: Variables from the Absenteeism at Work Data.

Variable Name	Data Type	Variable Name	Data Type	Range
Transportation expense	continuous	Number of children	ordinal	0,1,2,3
Distance from Residence	continuous	Number of pets	ordinal	0,1,2
Service time	continuous	Social drinker	nominal	0,1
Age	continuous	Social smoker	nominal	0,1
Work load day	continuous	Disciplinary failure	nominal	0,1
Hit target	continuous	Education	nominal	0,1,2
Weight	continuous	Seasons	nominal	0,1,2,3
Height	continuous	Day of the week	nominal	$0, 1, \cdots, 4$
Body mass index	continuous	Month of absence	nominal	$0, 1, \cdots, 11$
		Reason for absence	nominal	$0, 1, \cdots, 15$
		Absenteeism time	nominal	0,1

Breast Cancer Data: The dataset is observed or measured for 64 patients with breast cancer and 52 healthy controls. A binary variable indicates the presence or absence of breast cancer. There are 9 continuous variables from anthropometric data and parameters which can be gathered in routine blood analysis, including age, BMI, glucose, insulin, leptin and adiponectin, etc. We use 9 variables to predict the label of presence of breast cancer.

For classification problems, all continuous variables of the three datasets are normalized to get new original datasets. We consider different privacy budgets $\epsilon \in \{0.1, 0.5, 1, 1.5, \cdots, 5\}$ with a similar privacy budget split as in Section 4. Based on the results in Section 4, we consider the average division of privacy budget for the proposed method. For the DPRP method, we fix a small $\delta = n^{-5}$ to be comparable. We set the threshold $c = 0.8$ to select the number of factors. The average results based on 100 experiments are illustrated in Figures 5 - 7. The corresponding results with $\epsilon = 0.1$ and 5 for the three real data analyses are provided in the supplementary materials. From Figures 5 - 7, we observe that the proposed method outperforms the comparison methods for three real data analyses. Specifically, the proposed method has higher accuracy, F1 scores and AUC than the other methods. Moreover, the performance of the proposed method based on three criteria is robust for the privacy budget. In contrast, the performances of the four comparison methods fluctuate, especially on raw validating data. It implies that the proposed method can generate synthetic data with much higher utility for classification problems.

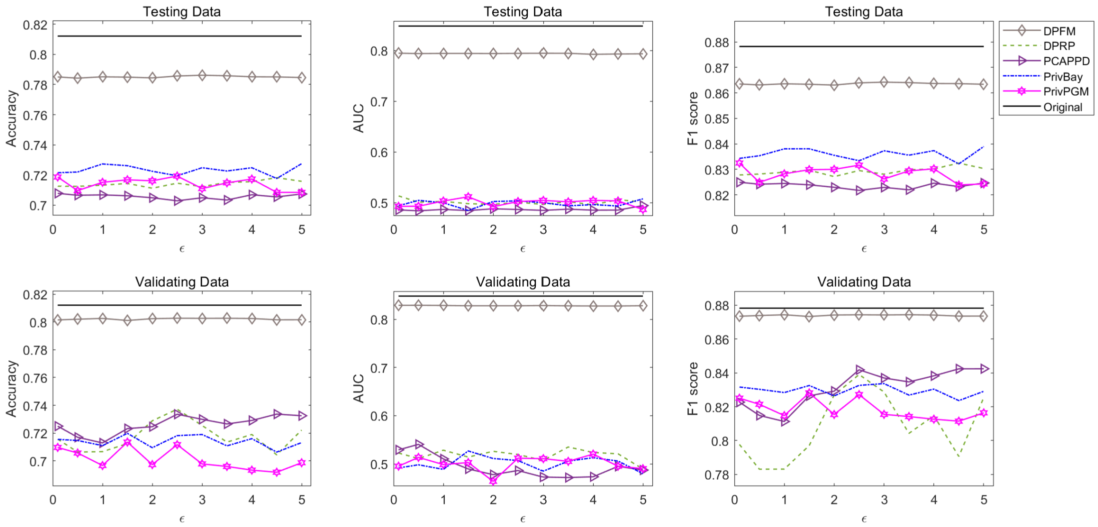

Figure 5: Results of three classifiers on testing synthetic data from five algorithms and validating data for the Census Income Data.

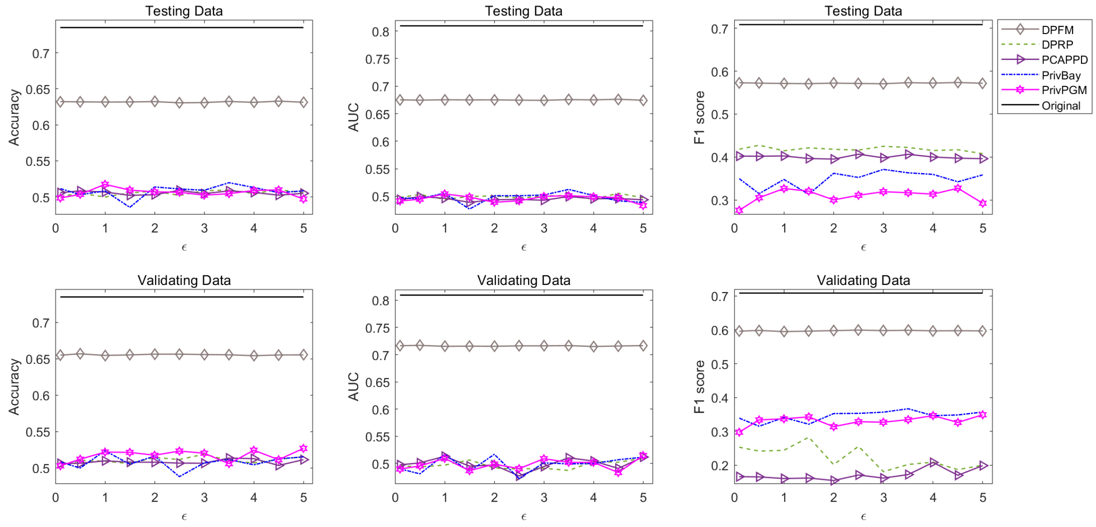

Figure 6: Results of three classifiers on testing synthetic data from five algorithms and validating data for the Absenteeism at Work Data.

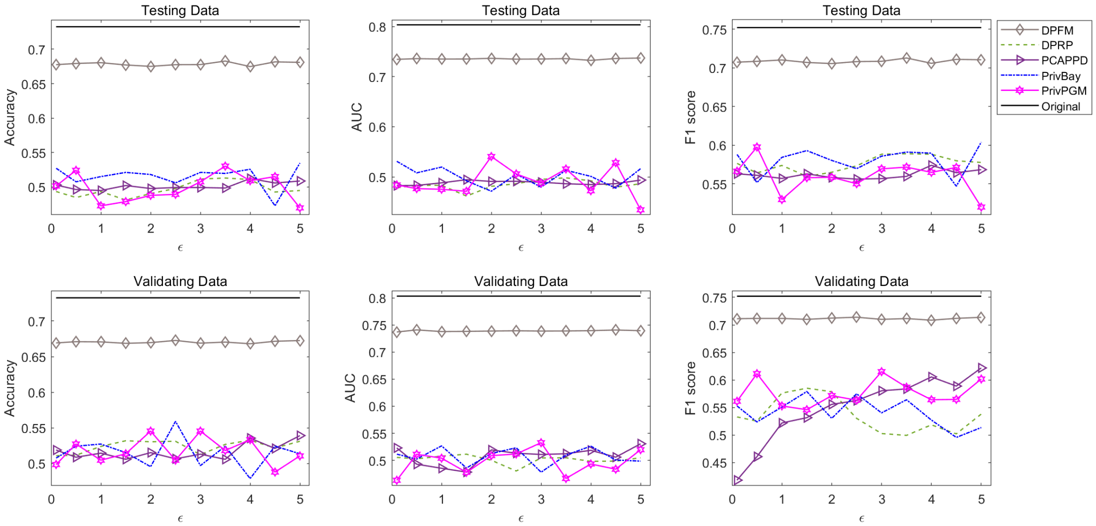

Figure 7: Results of three classifiers on testing synthetic data from five algorithms and validating data for the Breast Cancer Data.

6. Conclusion

In this paper we propose a novel algorithm for differential privacy synthetic data which achieves ϵ-differential privacy and provides synthetic data with a high utility guarantee. The proposed method adapts a factor model to preserve the main information of the correlated variables in generating synthetic data. One unique feature of our method is to utilize perturbation on the factor matrix and assign weighted privacy budgets to perturb eigenvectors associated with the original data based on the Laplace mechanism. Therefore, the proposed method maintains a low level of noise while maintaining the same level of privacy protection. Through continuous transformations with privacy protection, we can also deal with mixed-type data including categorical data and continuous data. Our extensive numerical studies indicate that the proposed differential privacy data-releasing algorithm is more effective than state-of-the art counterparts in maintaining data utility for highly correlated mixed-type data.

Acknowledgments and Disclosure of Funding

We would like to acknowledge support for this project from National Key R&D Program of China (102022YFA1003701), USA NSF grants (DMS1952406 and DMS2210640), NSF grants of China (12271472, 12231017 and 12001479), and NSF grants of Yunnan Province (202101AU070073 and 202201AT070101).

Appendix.

In this appendix we prove the following theorem from Section 3:

A. Proof of Lemma 5

Proof. Let $\boldsymbol{X}$ and $\boldsymbol{X}'$ contain at most one different record. Since $\boldsymbol{\mu}_i(\boldsymbol{X})$ and $\boldsymbol{\mu}_i(\boldsymbol{X}')$ are the i-th eigenvectors, we have

$$\|\boldsymbol{\mu}_i(\boldsymbol{X}) - \boldsymbol{\mu}_i(\boldsymbol{X}')\|_1 \leq \sqrt{p}\|\boldsymbol{\mu}_i(\boldsymbol{X}) - \boldsymbol{\mu}_i(\boldsymbol{X}')\|_2 \leq \sqrt{p}\{\|\boldsymbol{\mu}_i(\boldsymbol{X})\|_2 + \|\boldsymbol{\mu}_i(\boldsymbol{X}')\|_2\} \leq 2\sqrt{p}.$$

Thus, we obtain that

$$\Delta\boldsymbol{\mu}_i = \max_{\boldsymbol{X}':|\boldsymbol{X}-\boldsymbol{X}'|_1=1} \|\boldsymbol{\mu}_i(\boldsymbol{X}) - \boldsymbol{\mu}_i(\boldsymbol{X}')\|_1 \leq 2\sqrt{p}.$$

The proof of the lemma is completed.

B. Proof of Lemma 6

Proof. Consider any matrices $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ and $\boldsymbol{X}' \in \mathbb{R}^{n\times p}$ which contain at most one different record. Without loss of generality, let $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n-1}, \boldsymbol{x}_n)^\top$ and $\boldsymbol{X}' = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n-1}, \boldsymbol{x}'_n)^\top$, where $\boldsymbol{x}_n \neq \boldsymbol{x}'_n$, $\|\boldsymbol{x}'_n\|_2 \leq 1$, and $\|\boldsymbol{x}_i\|_2 \leq 1$ $(i = 1, \cdots, n)$. Since $\mathbf{T} = (\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_r)$ is a fixed matrix and $\boldsymbol{\mu}_k$ is a fixed unit vector, based on the Cauchy-Bunyakovsky-Schwarz inequality we can obtain that

$$\begin{array}{rcl}\|\mathbf{W}(\boldsymbol{X}) - \mathbf{W}(\boldsymbol{X}')\|_1 & = & \|\boldsymbol{X}\mathbf{T} - \boldsymbol{X}'\mathbf{T}\|_1 = \|(\boldsymbol{X} - \boldsymbol{X}')\mathbf{T}\|_1 = \|(\boldsymbol{x}_n - \boldsymbol{x}'_n)^\top\mathbf{T}\|_1 \\ & = & \sum_{k=1}^r |\sum_{j=1}^p (x_{nj} - x'_{nj})\boldsymbol{\mu}_{kj}| \\ & \leq & \sum_{k=1}^r \sum_{j=1}^p |(x_{nj} - x'_{nj})\boldsymbol{\mu}_{kj}| \\ & = & \sum_{k=1}^r \|(\boldsymbol{x}_n - \boldsymbol{x}'_n)^\top\boldsymbol{\mu}_k\|_1 \\ & \leq & \sum_{k=1}^r \|\boldsymbol{x}_n - \boldsymbol{x}'_n\|_2 \cdot \|\boldsymbol{\mu}_k\|_2 \leq 2r,\end{array}$$

where the norm of the matrix $\boldsymbol{A} \in \mathbb{R}^{n\times p}$ is $\|\boldsymbol{A}\|_1 = \sum_{i=1}^n \sum_{j=1}^p |a_{ij}|$. Thus, we have the ℓ_1-sensitivity is

$$\Delta\mathbf{W} = \max_{\boldsymbol{X}':|\boldsymbol{X}-\boldsymbol{X}'|_1=1} \|\mathbf{W}(\boldsymbol{X}) - \mathbf{W}(\boldsymbol{X}')\|_1 \leq 2r.$$

The proof of the lemma is completed.

C. Proof of Lemma 7

Proof. Consider any two vectors $\mathbf{z}_j \in \mathbb{R}^n$ and $\mathbf{z}'_j \in \mathbb{R}^n$ which contain at most one different record. Without loss of generality, let $\boldsymbol{z}_j = (z_{1j}, \cdots, z_{n-1,j}, z_{nj})^\top$ and $\mathbf{z}'_j = (z_{1j}, \cdots, z_{n-1,j}, z'_{nj})^\top$. Since $\boldsymbol{f}_j(\boldsymbol{z}_j) = (f_{j0}(\boldsymbol{z}_j), f_{j1}(\boldsymbol{z}_j), \cdots, f_{j,L_j-1}(\boldsymbol{z}_j), f_{j,L_j}(\boldsymbol{z}_j))$, $f_{jl}(\boldsymbol{x}_j) = \sum_{i=1}^n I(x_{ij} < l)/n$ and $0 = f_{j0}(\boldsymbol{x}_j) \leq f_{j1}(\boldsymbol{x}_j) \leq \cdots \leq f_{j,L_j-1}(\boldsymbol{x}_j) \leq f_{j,L_j}(\boldsymbol{x}_j) = 1$ for any $\boldsymbol{x}_j \in \mathbb{R}^n$, we have

$$\begin{array}{rcl}\|\boldsymbol{f}_j(\mathbf{z}_j) - \boldsymbol{f}_j(\mathbf{z}'_j)\|_1 & = & \sum_{k=1}^{L_j-1} |f_{jk}(\mathbf{z}_j) - f_{jk}(\mathbf{z}'_j)| \\ & = & \sum_{k=1}^{L_j-1} |I(z_{jn} < k) - I(z'_{jn} < k)|/n \\ & \leq & \sum_{k=1}^{L_j-1} 1/n \\ & = & (L_j - 1)/n.\end{array}$$

The proof of the lemma is completed.

D. Proof of Lemma 8

Proof. Let two $n \times p$ matrices $\boldsymbol{X}$ and $\boldsymbol{X}'$ contain at most one different record. Denote the perturbed eigenvectors $\widetilde{\boldsymbol{\mu}}_i = \boldsymbol{\mu}_i(\boldsymbol{X}) + \mathbf{b}$ and $\widetilde{\boldsymbol{\mu}}_i' = \boldsymbol{\mu}_i(\boldsymbol{X}') + \mathbf{b}'$, where each element of the vectors $\mathbf{b}$ and $\mathbf{b}' \in \mathbb{R}^p$ is from the Laplace distribution $Lap(2\sqrt{p}/\epsilon_{1i})$. Let $\boldsymbol{\mu}^* = (\mu_1^*, \cdots, \mu_p^*)^\top \in \mathbb{R}^p$. Based on Lemma 5, we can obtain that

$$\begin{aligned}
\frac{\Pr(\widetilde{\boldsymbol{\mu}}_i = \boldsymbol{\mu}^*|\boldsymbol{X})}{\Pr(\widetilde{\boldsymbol{\mu}}_i' = \boldsymbol{\mu}^*|\boldsymbol{X}')} &= \frac{\Pr(\mathbf{b} = \boldsymbol{\mu}^* - \boldsymbol{\mu}_i(\boldsymbol{X})|\boldsymbol{X})}{\Pr(\mathbf{b}' = \boldsymbol{\mu}^* - \boldsymbol{\mu}_i(\boldsymbol{X}')|\boldsymbol{X}')} \\
&= \exp\left\{\tfrac{\epsilon_{1i}}{2\sqrt{p}}(\|\boldsymbol{\mu}^* - \boldsymbol{\mu}_i(\boldsymbol{X}')\|_1 - \|\boldsymbol{\mu}^* - \boldsymbol{\mu}_i(\boldsymbol{X})\|_1)\right\} \\
&\leq \exp\left\{\tfrac{\epsilon_{1i}}{2\sqrt{p}}\|\boldsymbol{\mu}_i(\boldsymbol{X}') - \boldsymbol{\mu}_i(\boldsymbol{X})\|_1\right\} \\
&\leq \exp\left\{\tfrac{\epsilon_{1i}}{2\sqrt{p}} \max_{\boldsymbol{X}':\|\boldsymbol{X}-\boldsymbol{X}'\|_1=1} \|\boldsymbol{\mu}_i(\boldsymbol{X}') - \boldsymbol{\mu}_i(\boldsymbol{X})\|_1\right\} \\
&\leq \exp(\epsilon_{1i}).
\end{aligned}$$

Thus we have $\Pr(\widetilde{\boldsymbol{\mu}}_i = \boldsymbol{\mu}^*|\boldsymbol{X}) \leq e^{\epsilon_{1i}} \Pr(\widetilde{\boldsymbol{\mu}}_i' = \boldsymbol{\mu}^*|\boldsymbol{X}')$ for any $\boldsymbol{\mu}^* \in \mathbb{R}^p$. Based on the definition of ϵ-differential privacy (Dwork and Roth, 2014), the lemma holds.

E. Proof of Lemma 9

Proof. Let $\boldsymbol{V}(\boldsymbol{X}) = (\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_r)$, where $\boldsymbol{\mu}_i$ is the eigenvector corresponding to the i-th largest eigenvalue of the matrix $\boldsymbol{X}^\top \boldsymbol{X}$. We know that

$$\mathcal{M}(\boldsymbol{X}) = \boldsymbol{V}(\boldsymbol{X}) + \mathbf{B} = (\boldsymbol{\mu}_1 + \mathbf{b}_1, \cdots, \boldsymbol{\mu}_r + \mathbf{b}_r).$$

Based on Lemma 8 and composition theorem (Dwork and Roth, 2014), we can obtain that $\mathcal{M}$ is $\sum_{i=1}^r \epsilon_{1i}$-differentially private, that is, ϵ_1-differentially private. The proof of the lemma is completed.

F. Proof of Theorem 10

Proof. Consider any matrices $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ and $\boldsymbol{X}' \in \mathbb{R}^{n\times p}$ which contain at most one different record. That is, let $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n-1}, \boldsymbol{x}_n)^\top$ and $\boldsymbol{X}' = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n-1}, \boldsymbol{x}_n')^\top$, where $\boldsymbol{x}_n \neq \boldsymbol{x}_n'$, $\|\boldsymbol{x}_n'\|_2 \leq 1$, and $\|\boldsymbol{x}_i\|_2 \leq 1$ $(i = 1, \cdots, n)$. Denote $\widetilde{\boldsymbol{X}}$ and $\widetilde{\boldsymbol{X}'}$ as the matrices released via Algorithm 1 based on $\boldsymbol{X}$ and $\boldsymbol{X}'$, respectively. That is, $\widetilde{\boldsymbol{X}} = \widetilde{\mathbf{W}}\widetilde{\boldsymbol{\Lambda}}^\top$, and $\widetilde{\boldsymbol{X}'} = \widetilde{\mathbf{W}'}\widetilde{\boldsymbol{\Lambda}'}^\top$, where $\widetilde{\boldsymbol{\Lambda}}$ and $\widetilde{\boldsymbol{\Lambda}'}$ are the perturbed factor loading matrices from Step 3(II) of Algorithm1 based on $\boldsymbol{X}$ and $\boldsymbol{X}'$, respectively, and $\widetilde{\mathbf{W}} = \boldsymbol{X}\widetilde{\boldsymbol{\Lambda}} + \boldsymbol{C}$ and $\widetilde{\mathbf{W}'} = \boldsymbol{X}'\widetilde{\boldsymbol{\Lambda}'} + \boldsymbol{C}'$ are the perturbed factor matrices, in which $\boldsymbol{C}$ and $\boldsymbol{C}'$ are generated from the Step 3(I) of Algorithm 1. Denote $\mathbb{Z}$ as the range of the synthesized matrix released via Algorithm 1.

For any $\mathbf{Z} \in \mathbb{Z}$, the privacy loss satisfies that

$$
\frac{\Pr(\widetilde{\boldsymbol{X}} = \mathbf{Z}|\boldsymbol{X})}{\Pr(\widetilde{\boldsymbol{X}'} = \mathbf{Z}|\boldsymbol{X}')} = \frac{\displaystyle\int_{\mathbb{T}}\int_{\mathbb{N}} \Pr(\widetilde{\mathbf{W}}\widetilde{\boldsymbol{\Lambda}}^{\top} = \mathbf{Z}, \widetilde{\mathbf{W}} = \mathbf{N}, \widetilde{\boldsymbol{\Lambda}} = \mathbf{T}|\boldsymbol{X}) d\mathbf{N} d\mathbf{T}}{\displaystyle\int_{\mathbb{T}}\int_{\mathbb{N}} \Pr(\widetilde{\mathbf{W}'}\widetilde{\boldsymbol{\Lambda}'}^{\top} = \mathbf{Z}, \widetilde{\mathbf{W}'} = \mathbf{N}, \widetilde{\boldsymbol{\Lambda}'} = \mathbf{T}|\boldsymbol{X}') d\mathbf{N} d\mathbf{T}} =
$$

$$
\frac{\displaystyle\int_{\mathbb{T}}\int_{\mathbb{N}} \Pr\{\widetilde{\mathbf{W}}\widetilde{\boldsymbol{\Lambda}}^{\top} = \mathbf{Z}|\widetilde{\mathbf{W}} = \mathbf{N}, \widetilde{\boldsymbol{\Lambda}} = \mathbf{T}\} \Pr(\boldsymbol{X}\widetilde{\boldsymbol{\Lambda}} + \boldsymbol{C} = \mathbf{N}|\boldsymbol{X}, \widetilde{\boldsymbol{\Lambda}} = \mathbf{T}) \Pr(\widetilde{\boldsymbol{\Lambda}} = \mathbf{T}|\boldsymbol{X}) d\mathbf{N} d\mathbf{T}}{\displaystyle\int_{\mathbb{T}}\int_{\mathbb{N}} \Pr\{\widetilde{\mathbf{W}'}\widetilde{\boldsymbol{\Lambda}'}^{\top} = \mathbf{Z}|\widetilde{\mathbf{W}'} = \mathbf{N}, \widetilde{\boldsymbol{\Lambda}'} = \mathbf{T}\} \Pr(\boldsymbol{X}'\widetilde{\boldsymbol{\Lambda}'} + \boldsymbol{C}' = \mathbf{N}|\boldsymbol{X}', \widetilde{\boldsymbol{\Lambda}'} = \mathbf{T}) \Pr(\widetilde{\boldsymbol{\Lambda}'} = \mathbf{T}|\boldsymbol{X}') d\mathbf{N} d\mathbf{T}} \tag{10}
$$

where $\mathbb{N}$ and $\mathbb{T}$ are the ranges of the factor matrix and loading matrix from the Step 3(II) of Algorithm 1. Based on Step 2(ii) and Step 3(II) of Algorithm 1, we know that $\mathbb{T}$ consists of $p \times r$ unit orthogonal matrices. Note that the above integrals represent the multiple integrals with respect to each entry in the matrix $\mathbf{N}$ and $\mathbf{T}$.

Based on Lemma 9, we can show that $\boldsymbol{V}^*$ generated via the Step 2(i) of Algorithm 1 is ϵ_1-differentially private. Based on Step 2(ii) in the Algorithm 1, $\widetilde{\boldsymbol{V}} = \boldsymbol{g}(\boldsymbol{V}^*)$, where $\boldsymbol{g}(\boldsymbol{V}^*)$ is a $p \times r$ unit orthogonal matrix obtained by the top r left-singular vectors of the matrix $\boldsymbol{V}^*$. That is, $\boldsymbol{V}^* = \boldsymbol{SQR}$ and $\widetilde{\boldsymbol{V}} = \boldsymbol{S}_r$, where $\boldsymbol{Q}$ is a $p \times r$ rectangular diagonal matrix with singular values of $\boldsymbol{V}^*$ in decreasing order on the diagonal, $\boldsymbol{R}$ is a $r \times r$ unit orthogonal matrix with each column being the corresponding right-singular vector, $\boldsymbol{S}$ is a $p \times p$ unit orthogonal matrix with each column being the corresponding left-singular vector, and $\boldsymbol{S}_r$ is a $p \times r$ unit orthogonal matrix with the columns being the top r left-singular vectors. Based on the post-processing property of differential privacy (Dwork and Roth, 2014), $\widetilde{\boldsymbol{V}}$ from Step 2(ii) of Algorithm 1 is also ϵ_1-differentially private. Based on the definition of ϵ-differential privacy (Dwork and Roth, 2014), we have that $\Pr(\widetilde{\boldsymbol{V}} = \mathbf{T}|\boldsymbol{X}) \leq e^{\epsilon_1} \Pr(\widetilde{\boldsymbol{V}'} = \mathbf{T}|\boldsymbol{X}')$ for any $\mathbf{T} \in \mathbb{T}$. Thus, we have

$$
\Pr(\widetilde{\boldsymbol{\Lambda}} = \mathbf{T}|\boldsymbol{X}) \leq e^{\epsilon_1} \Pr(\widetilde{\boldsymbol{\Lambda}'} = \mathbf{T}|\boldsymbol{X}'). \tag{11}
$$

Based on Lemma 6, we have

$$
\begin{aligned}
\frac{\Pr\left(\boldsymbol{X}\widetilde{\boldsymbol{\Lambda}} + \boldsymbol{C} = \mathbf{N}|\boldsymbol{X}, \widetilde{\boldsymbol{\Lambda}} = \mathbf{T}\right)}{\Pr\left(\boldsymbol{X}'\widetilde{\boldsymbol{\Lambda}'} + \boldsymbol{C}' = \mathbf{N}|\boldsymbol{X}', \widetilde{\boldsymbol{\Lambda}'} = \mathbf{T}\right)} &= \frac{\Pr\left(\boldsymbol{C} = \mathbf{N} - \boldsymbol{X}\widetilde{\boldsymbol{\Lambda}}|\boldsymbol{X}, \widetilde{\boldsymbol{\Lambda}} = \mathbf{T}\right)}{\Pr\left(\boldsymbol{C}' = \mathbf{N} - \boldsymbol{X}'\widetilde{\boldsymbol{\Lambda}'}|\boldsymbol{X}', \widetilde{\boldsymbol{\Lambda}'} = \mathbf{T}\right)} \\
&= \frac{\Pr\left(\boldsymbol{C} = \mathbf{N} - \boldsymbol{X}\mathbf{T}|\boldsymbol{X}, \widetilde{\boldsymbol{\Lambda}} = \mathbf{T}\right)}{\Pr\left(\boldsymbol{C}' = \mathbf{N} - \boldsymbol{X}'\mathbf{T}|\boldsymbol{X}', \widetilde{\boldsymbol{\Lambda}'} = \mathbf{T}\right)} \\
&= \exp\left\{\tfrac{\epsilon_2}{2r}(\|\mathbf{N} - \boldsymbol{X}'\mathbf{T}\|_1 - \|\mathbf{N} - \boldsymbol{X}\mathbf{T}\|_1)\right\} \\
&\leq \exp\left\{\tfrac{\epsilon_2}{2r}\|(\boldsymbol{X} - \boldsymbol{X}')\mathbf{T}\|_1\right\} \leq e^{\epsilon_2}.
\end{aligned}
$$

Therefore we have

$$
\Pr(\boldsymbol{X}\widetilde{\boldsymbol{\Lambda}} + \boldsymbol{C} = \mathbf{N}|\boldsymbol{X}, \widetilde{\boldsymbol{\Lambda}} = \mathbf{T}) \leq e^{\epsilon_2} \Pr(\boldsymbol{X}'\widetilde{\boldsymbol{\Lambda}'} + \boldsymbol{C}' = \mathbf{N}|\boldsymbol{X}', \widetilde{\boldsymbol{\Lambda}'} = \mathbf{T}). \tag{12}
$$

Based on the inequality (11)-(12) and the equality (10), we have

$$\frac{\Pr(\widetilde{\boldsymbol{X}} = \mathbf{Z}|\boldsymbol{X})}{\Pr(\widetilde{\boldsymbol{X}'} = \mathbf{Z}|\boldsymbol{X}')} \leq$$
$$\frac{\int_{\mathbb{T}}\int_{\mathbb{N}} \Pr\{\widetilde{\mathbf{W}}\widetilde{\widetilde{\boldsymbol{\Lambda}}}^{\top} = \mathbf{Z}|\widetilde{\mathbf{W}} = \mathbf{N}, \widetilde{\boldsymbol{\Lambda}} = \mathbf{T}\}e^{\epsilon_2} \Pr(\boldsymbol{X}'\widetilde{\boldsymbol{\Lambda}'} + \boldsymbol{C}' = \mathbf{N}|\boldsymbol{X}', \widetilde{\boldsymbol{\Lambda}'} = \mathbf{T})e^{\epsilon_1} \Pr(\widetilde{\boldsymbol{\Lambda}'} = \mathbf{T}|\boldsymbol{X}')d\mathbf{N}d\mathbf{T}}{\int_{\mathbb{T}}\int_{\mathbb{N}} \Pr\{\widetilde{\mathbf{W}'}\widetilde{\boldsymbol{\Lambda}'}^{\top} = \mathbf{Z}|\widetilde{\mathbf{W}'} = \mathbf{N}, \widetilde{\boldsymbol{\Lambda}'} = \mathbf{T}\} \Pr(\boldsymbol{X}'\widetilde{\boldsymbol{\Lambda}'} + \boldsymbol{C}' = \mathbf{N}|\boldsymbol{X}', \widetilde{\boldsymbol{\Lambda}'} = \mathbf{T}) \Pr(\widetilde{\boldsymbol{\Lambda}'} = \mathbf{T}|\boldsymbol{X}')d\mathbf{N}d\mathbf{T}}$$
$$= e^{\epsilon_1+\epsilon_2}.$$

That is, the privacy loss satisfies

$$\max_{\mathbf{Z}\in\mathbb{Z}} \frac{\Pr(\widetilde{\boldsymbol{X}} = \mathbf{Z}|\boldsymbol{X})}{\Pr(\widetilde{\boldsymbol{X}'} = \mathbf{Z}|\boldsymbol{X}')} \leq e^{\epsilon_1+\epsilon_2}.$$

Thus, Algorithm 1 is $\epsilon_1 + \epsilon_2$-differentially private. The proof of the theorem is completed.

Lemma 16 *For any matrices $\boldsymbol{A} \in \mathbb{R}^{n\times m}$ and $\boldsymbol{D} \in \mathbb{R}^{n\times m}$ where the rank of $\boldsymbol{A}$ is p and $p = \min(m, n)$,*

$$\|\boldsymbol{\Pi}_A^{(p)}(I - \boldsymbol{\Pi}_D^{(p)})\|_2 \leq \|\boldsymbol{A} - \boldsymbol{D}\|_2/\sigma_p(\boldsymbol{A}),$$

where $\boldsymbol{\Pi}_A^k$ is the projector to the subspace spanned by the top k left singular vectors of the matrix $\boldsymbol{A}$, and $\boldsymbol{\Pi}_D^k$ is also the projector. That is, $\boldsymbol{\Pi}_A^{(p)} = \boldsymbol{U}\boldsymbol{U}^\top$, $\boldsymbol{U}$ is a matrix with the columns being the first p left singular vectors of $\boldsymbol{A}$, and $\sigma_p(\boldsymbol{A})$ is the p-th singular value.

Proof. The lemma can be prived based on the proof of the spectral perturbation bound from matrix perturbation theory (Mcsherry and Karlin, 2004). This proof is inspired by the proof of the theorem 7 in (Mcsherry and Karlin, 2004). We first start by proving the bound for symmetric matrices $\boldsymbol{A}$ and $\boldsymbol{D}$, where $\boldsymbol{\Pi}_A^{(p)} = \boldsymbol{\Pi}_{A^\top}^{(p)}$, and $\boldsymbol{\Pi}_D^{(p)} = \boldsymbol{\Pi}_{D^\top}^{(p)}$, in which $\boldsymbol{\Pi}_{A^\top}^{(p)} = \boldsymbol{V}\boldsymbol{V}^\top$ and $\boldsymbol{V}$ is a matrix with the columns being the first p right singular vectors of $\boldsymbol{A}$. At this point, $n = m = p$. Notice that in this case

$$\begin{aligned}
\boldsymbol{\Pi}_A^{(p)}(\boldsymbol{A} - \boldsymbol{D})(\mathrm{I} - \boldsymbol{\Pi}_D^{(p)}) &= \boldsymbol{\Pi}_A^{(p)}\boldsymbol{A}(\mathrm{I} - \boldsymbol{\Pi}_D^{(p)}) - \boldsymbol{\Pi}_A^{(p)}\boldsymbol{D}(\mathrm{I} - \boldsymbol{\Pi}_D^{(p)}) \\
&= \boldsymbol{A}\boldsymbol{\Pi}_{A^\top}^{(p)}(\mathrm{I} - \boldsymbol{\Pi}_D^{(p)}) - \boldsymbol{\Pi}_A^{(p)}\boldsymbol{D}(\mathrm{I} - \boldsymbol{\Pi}_{D^\top}^{(p)}) \\
&= \boldsymbol{A}\boldsymbol{\Pi}_A^{(p)}(\mathrm{I} - \boldsymbol{\Pi}_D^{(p)}) - \boldsymbol{\Pi}_A^{(p)}(\mathrm{I} - \boldsymbol{\Pi}_D^{(p)})\boldsymbol{D}.
\end{aligned}$$

Recall from its definition that the L_2 norm is defined by a unit vector which undergoes maximum stretch when multiplied by the matrix. Let x be an unit vector such that $|\boldsymbol{\Pi}_A^{(p)}(\mathrm{I} - \boldsymbol{\Pi}_D^{(p)})x| = \|\boldsymbol{\Pi}_A^{(p)}(\mathrm{I} - \boldsymbol{\Pi}_D^{(p)})\|_2$. Based on the triangle inequality, we have

$$|\boldsymbol{\Pi}_A^{(p)}(\boldsymbol{A} - \boldsymbol{D})(\mathrm{I} - \boldsymbol{\Pi}_D^{(p)})x| \geq |\boldsymbol{A}\boldsymbol{\Pi}_A^{(p)}(\mathrm{I} - \boldsymbol{\Pi}_D^{(p)})x| - |\boldsymbol{\Pi}_A^{(p)}(\mathrm{I} - \boldsymbol{\Pi}_D^{(p)})\boldsymbol{D}x|.$$

Note that $\boldsymbol{\Pi}_A^{(p)}(\mathrm{I} - \boldsymbol{\Pi}_D^{(p)})x$ lies in the space spanned by the first p left singular vectors of $\boldsymbol{A}$, and so when multiplied by $\boldsymbol{A}$ its norm increases by at least a factor of $\sigma_p(\boldsymbol{A})$. And $\boldsymbol{\Pi}_A^{(p)}(\mathrm{I} - \boldsymbol{\Pi}_D^{(p)})$ annihilates any aspect of $\boldsymbol{D}x$ that emerges on the first p left singular vectors of $\boldsymbol{D}$, and so $|\mathbf{\Pi}_A^{(p)}(\mathrm{I}-\mathbf{\Pi}_D^{(p)})\boldsymbol{D}x| = |\mathbf{\Pi}_A^{(p)}(\mathrm{I}-\mathbf{\Pi}_D^{(p)})(\boldsymbol{D}-\boldsymbol{D}^{(p)})x| = 0$, where $\boldsymbol{D}^{(p)} = \sum_{i\leq p}\sigma_i\boldsymbol{u}_i\boldsymbol{v}_i^\top = \boldsymbol{D}$, σ_i is the i-th singular value of $\boldsymbol{D}$, and $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ are the corresponding left and right singular vectors of $\boldsymbol{D}$, respectively. Thus, we can obtain that

$$\|\boldsymbol{A}-\boldsymbol{D}\|_2 \geq \sigma_p(\boldsymbol{A})\|\mathbf{\Pi}_A^{(p)}(\mathrm{I}-\mathbf{\Pi}_D^{(p)})\|_2.$$

Then, we have

$$\|\mathbf{\Pi}_A^{(p)}(\mathrm{I}-\mathbf{\Pi}_D^{(p)})\|_2 \leq \|\boldsymbol{A}-\boldsymbol{D}\|_2/\sigma_p(\boldsymbol{A}).$$

The proof for the case of symmetric $\boldsymbol{A}$ and $\boldsymbol{D}$ is completed. Based on the above result, the proof for the case of arbitrary matrices can be completed simularly to the proof of the theorem 7 in Mcsherry and Karlin (2004).

G. Proof of Lemma 11

Proof. Let $\mathbf{\Pi} = \boldsymbol{V}\boldsymbol{V}^\top$ and $\widetilde{\mathbf{\Pi}} = \widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{V}}^\top$. It is easy to obtain the following equation

$$\mathbf{\Pi}-\widetilde{\mathbf{\Pi}} = \mathbf{\Pi}-\mathbf{\Pi}\widetilde{\mathbf{\Pi}}+\mathbf{\Pi}\widetilde{\mathbf{\Pi}}-\widetilde{\mathbf{\Pi}} = \mathbf{\Pi}(\mathrm{I}-\widetilde{\mathbf{\Pi}})-(\mathrm{I}-\mathbf{\Pi})\widetilde{\mathbf{\Pi}}.$$

Based on the inequality that $\|\boldsymbol{A}\|_F \leq \sqrt{r}\|\boldsymbol{A}\|_2$ where r is the rank of the matrix $\boldsymbol{A}$ and $\|\boldsymbol{A}\|_F$ and $\|\boldsymbol{A}\|_2$ are the Frobenius norm and the spectral norm respectively, we have

$$\begin{aligned}\|\mathbf{\Pi}-\widetilde{\mathbf{\Pi}}\|_F &= \|\mathbf{\Pi}(\mathrm{I}-\widetilde{\mathbf{\Pi}})-(\mathrm{I}-\mathbf{\Pi})\widetilde{\mathbf{\Pi}}\|_F\\ &\leq \|\mathbf{\Pi}(\mathrm{I}-\widetilde{\mathbf{\Pi}})\|_F+\|(\mathrm{I}-\mathbf{\Pi})\widetilde{\mathbf{\Pi}}\|_F\\ &= \|\mathbf{\Pi}(\mathrm{I}-\widetilde{\mathbf{\Pi}})\|_F+\|\widetilde{\mathbf{\Pi}}(\mathrm{I}-\mathbf{\Pi})\|_F\\ &\leq \sqrt{r}(\|\mathbf{\Pi}(\mathrm{I}-\widetilde{\mathbf{\Pi}})\|_2+\|\widetilde{\mathbf{\Pi}}(\mathrm{I}-\mathbf{\Pi})\|_2).\end{aligned}$$

Let $\boldsymbol{A} = \boldsymbol{V}$ and $\boldsymbol{D} = \boldsymbol{V}+\mathbf{B}$. Since the $\boldsymbol{V}$ is a $p\times r$ unit orthogonal matrix with the eignvectors as the columns, the singular values of the $\boldsymbol{A}$ are $\sigma_j(\boldsymbol{A}) = 1$ for $j = 1, 2, \ldots, r$, and the projector to the subspace spanned by the top r singular vectors of the $\boldsymbol{A}$ is $\mathbf{\Pi}_r(\boldsymbol{A}) = \boldsymbol{V}\boldsymbol{V}^\top$. Based on Step 2(ii) in the Algorithm 1, it is easy to find that the projector to the subspace spanned by the top r singular vectors of the $\boldsymbol{D}$ is $\mathbf{\Pi}_r(\boldsymbol{D}) = \widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{V}}^\top$. Based on the above lemma 16, we have

$$\begin{aligned}\|\mathbf{\Pi}(\mathrm{I}-\widetilde{\mathbf{\Pi}})\|_2 &= \|\mathbf{\Pi}_r(\boldsymbol{A})(\mathrm{I}-\mathbf{\Pi}_r(\boldsymbol{D}))\|_2\\ &\leq \|\boldsymbol{A}-\boldsymbol{D}\|_2/\sigma_r(\boldsymbol{A}) = \|-\mathbf{B}\|_2.\end{aligned}$$

Since each entry of the j-th column of $\mathbf{B} = (\mathbf{b}_1\cdots, \mathbf{b}_r)$ is independent from $Lap(2\sqrt{p}/\epsilon_{1j})$, we have $E(\|\mathbf{B}\|_F^2) = \sum_{j=1}^r E(\|\mathbf{b}_j\|_2^2) = 8p^2\sum_{j=1}^r 1/\epsilon_{1j}^2$. Based on Jensen's inequality, we have $E(\|\mathbf{B}\|_F) \leq \sqrt{E(\|\mathbf{B}\|_F^2)} = 2\sqrt{2}p\sqrt{\sum_{j=1}^r 1/\epsilon_{1j}^2}$. Based on Markov's inequality, we have with high probability $\|\mathbf{B}\|_2 = O(p/\epsilon_1\sqrt{\sum_{j=1}^r 1/\omega_j^2})$. Thus, we have that

$$\|\mathbf{\Pi}(\mathrm{I}-\widetilde{\mathbf{\Pi}})\|_2 = O(p/\epsilon_1\sqrt{\textstyle\sum_{j=1}^r 1/\omega_j^2}).$$

Let $\boldsymbol{A} = \boldsymbol{V}+\boldsymbol{B}$ and $\boldsymbol{D} = \boldsymbol{V}$. Based on the above lemma 16, we have

$$\|\widetilde{\mathbf{\Pi}}(\mathrm{I}-\mathbf{\Pi})\|_2 \leq \|\boldsymbol{A}-\boldsymbol{D}\|_2/\sigma_r(\boldsymbol{A}) = \|\mathbf{B}\|_2/\sigma_r(\boldsymbol{A}).$$

since $\sigma_r(\boldsymbol{A}) = \sigma_r(\boldsymbol{V}+\boldsymbol{B}) \geq \sigma_r(\boldsymbol{V}) + \sigma_r(\boldsymbol{B}) = 1 + \sigma_r(\boldsymbol{B}) \geq 1$, we can obtain

$$\|\widetilde{\mathbf{\Pi}}(\mathrm{I}-\mathbf{\Pi})\|_2 \leq \|\mathbf{B}\|_2/\sigma_r(\boldsymbol{A}) \leq \|\mathbf{B}\|_2.$$

Similarly, we have that

$$\|\widetilde{\mathbf{\Pi}}(\mathrm{I}-\mathbf{\Pi})\|_2 = O(p/\epsilon_1\sqrt{\textstyle\sum_{j=1}^r 1/\omega_j^2}).$$

Since $\sum_{j=1}^r 1/\omega_j^2 = O(1)$, we can obtain that

$$\|\boldsymbol{V}\boldsymbol{V}^\top - \widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{V}}^\top\|_F = \|\mathbf{\Pi}-\widetilde{\mathbf{\Pi}}\|_F = O(\sqrt{r}p/\epsilon_1\sqrt{\textstyle\sum_{j=1}^r 1/\omega_j^2}) = O(\sqrt{r}p/\epsilon_1).$$

The proof of the lemma is completed.

H. Proof of Theorem 12

Proof. Based on the framework of the factor model, we have $\widehat{\boldsymbol{X}} = \widehat{\mathbf{W}}\widehat{\mathbf{\Lambda}}^\top = \boldsymbol{X}\boldsymbol{V}\boldsymbol{V}^\top$, where $\boldsymbol{V} = (\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_r)$ is a $p \times r$ matrix and $\boldsymbol{\mu}_i$ $(i = 1, \cdots, r)$ is the eigenvector corresponding to the i-th largest eigenvalue of the matrix $\boldsymbol{A} = \boldsymbol{X}^\top\boldsymbol{X}$ in decreasing order. Based on Algorithm 1, we have $\widetilde{\boldsymbol{X}} = \widetilde{\mathbf{W}}\widetilde{\mathbf{\Lambda}}^\top = (\boldsymbol{X}\widetilde{\boldsymbol{V}} + \boldsymbol{C})\widetilde{\boldsymbol{V}}^\top = \boldsymbol{X}\widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{V}}^\top + \boldsymbol{C}\widetilde{\boldsymbol{V}}^\top$. Thus, we can obtain

$$\widetilde{\boldsymbol{X}} - \boldsymbol{X} = \widehat{\boldsymbol{X}} - \boldsymbol{X} + \boldsymbol{X}(\widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{V}}^\top - \boldsymbol{V}\boldsymbol{V}^\top) + \boldsymbol{C}\widetilde{\boldsymbol{V}}^\top.$$

Since $\|\boldsymbol{x}_i\|_2 \leq 1$, we have $\|\boldsymbol{X}\|_F \leq \sqrt{n}$. Since $\widetilde{\boldsymbol{\mu}}_r$ is a unit vector, we can obtain that

$$\|\widetilde{\boldsymbol{X}} - \boldsymbol{X}\|_F \leq \|\widehat{\boldsymbol{X}} - \boldsymbol{X}\|_F + \sqrt{n}\|\widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{V}}^\top - \boldsymbol{V}\boldsymbol{V}^\top\|_F + \| + \boldsymbol{C}\widetilde{\boldsymbol{V}}^\top\|_F.$$

Based on Step 3(I) in Algorithm 1, we know that each entry of $\boldsymbol{C}$ is from $Lap(2r/\epsilon_2)$. Based on Jensen's inequality, we have

$$E\left(\|\boldsymbol{C}\|_F\right) = E\left(\sqrt{\textstyle\sum_{k=1}^r\sum_{j=1}^n c_{kj}^2}\right) \leq \sqrt{\textstyle\sum_{k=1}^r\sum_{j=1}^n E(c_{kj}^2)} = 2r\sqrt{2rn}/\epsilon_2.$$

Since $\boldsymbol{X}$ has eigenvalues $\nu_1, \cdots, \nu_{\min(n,p)}$, we have

$$\|\widehat{\boldsymbol{X}} - \boldsymbol{X}\|_F = \|\boldsymbol{X}\boldsymbol{V}\boldsymbol{V}^\top - \boldsymbol{X}\|_F = \sqrt{\sum_{i=r}^{\min(n,p)} \nu_i} < \sqrt{\sum_{i=1}^{\min(n,p)} \nu_i} = \|\boldsymbol{X}\|_F \leq \sqrt{n}.$$

Since $\epsilon_1 = c\epsilon$ and $\epsilon_2 = (1-c)\epsilon$ for $c \in (0,1)$ and the number r is fixed, based on Theorem 11 and Markov's inequality, we have with high probability

$$\begin{array}{rcl} \|\widetilde{\boldsymbol{X}} - \boldsymbol{X}\|_F & \leq & \|\widehat{\boldsymbol{X}} - \boldsymbol{X}\|_F + O(\sqrt{r}p/\epsilon_1 + r\sqrt{rn}/\epsilon_2) \\ & \leq & \|\widehat{\boldsymbol{X}} - \boldsymbol{X}\|_F + O((p+\sqrt{n})/\epsilon) \\ & \leq & O\left(\sqrt{n} + (p+\sqrt{n})/\epsilon\right). \end{array}$$

The proof of the theorem is completed.

I. Proof of Theorem 15

Proof. Let $\boldsymbol{X} = (\boldsymbol{Y}, \mathbf{Z}, \boldsymbol{U}) \in \mathbb{R}^{n\times p}$ be original data, where $\boldsymbol{Y} \in \mathbb{R}^{n\times p_1}$ is the continuous data matrix, $\mathbf{Z} \in \mathbb{R}^{n\times p_2}$ is the ordinal categorical data matrix, and $\boldsymbol{U} \in \mathbb{R}^{n\times p_3}$ is the nominal categorical data matrix. Let $\widetilde{\boldsymbol{X}} = (\widetilde{\boldsymbol{Y}}, \widetilde{\mathbf{Z}}, \widetilde{\boldsymbol{U}}) \in \mathbb{R}^{n\times p}$ be synthetic data from Algorithm 3. Then we have

$$\|\widetilde{\boldsymbol{X}} - \boldsymbol{X}\|_F^2 = \|\widetilde{\boldsymbol{Y}} - \boldsymbol{Y}\|_F^2 + \|\widetilde{\mathbf{Z}} - \mathbf{Z}\|_F^2 + \|\widetilde{\boldsymbol{U}} - \boldsymbol{U}\|_F^2,$$

where

$$\|\widetilde{\mathbf{Z}} - \mathbf{Z}\|_F^2 = \sum_{i=1}^{n}\sum_{j=1}^{p_2} |\widetilde{z}_{ij} - z_{ij}|^2 \leq n\sum_{j=1}^{p_2}(L_j - 1)^2,$$

and

$$\|\widetilde{\boldsymbol{U}} - \boldsymbol{U}\|_F^2 = \sum_{i=1}^{n}\sum_{j=1}^{p_3} |\widetilde{u}_{ij} - u_{ij}|^2 \leq n\sum_{j=1}^{p_3}(M_j - 1)^2.$$

Based on Theorem 12, we have

$$\|\widetilde{\boldsymbol{Y}} - \boldsymbol{Y}\|_F = O\left(\sqrt{n} + (p_1 + \sqrt{n})/\epsilon\right).$$

Since $\max_{1\leq j\leq p_2}(L_j - 1)^2 = O(1)$ and $\max_{1\leq j\leq p_3}(M_j - 1)^2 = O(1)$, we can obtain that

$$\begin{aligned} & \|\widetilde{\boldsymbol{X}} - \boldsymbol{X}\|_F \leq \|\widetilde{\boldsymbol{Y}} - \boldsymbol{Y}\|_F + \|\widetilde{\mathbf{Z}} - \mathbf{Z}\|_F + \|\widetilde{\boldsymbol{U}} - \boldsymbol{U}\|_F \\ = \; & O\left(\sqrt{n} + (p_1 + \sqrt{n})/\epsilon + \sqrt{n\textstyle\sum_{j=1}^{p_2}(L_j - 1)^2} + \sqrt{n\textstyle\sum_{j=1}^{p_3}(M_j - 1)^2}\right) \\ = \; & O\left(\sqrt{n} + (p_1 + \sqrt{n})/\epsilon + \sqrt{np_2} + \sqrt{np_3}\right). \end{aligned}$$

The proof of the theorem is completed.

J. Simulation results

The results of the simulation studies as $\epsilon = 0.1$ and 5 are reported in Tables 3 and 4. From Table 3, we find that when $p = 22$, the maximum discrepancies of the accuracy, AUC and F1 score of the proposed methods compared with those of the original data are 0.038, 0.058 and 0.020, respectively. However, the maximum discrepancies of the three criteria of the competing methods from those of the original data are 0.135, 0.544 and 0.081, respectively. From Table 4, we show that when $p = 80$, the maximum discrepancies of the three criteria of the proposed method from those of the original data are 0.065, 0.120 and 0.034, respectively. However, the maximum discrepancies of the three criteria of the competing methods from those of the original data are 0.101, 0.540 and 0.060, respectively. This implies that although the proposed methods also have utility losses in order to satisfy certain privacy protection requirements, the utility losses are much lower than those of the competing methods in all of simulation settings.

Table 3: Average criteria of three classifiers and standard deviation in parentheses for synthetic testing data from eight algorithms and raw validating data in setting (I).

	Testing Data			Validating Data		
	accuracy	AUC	F1 score	accuracy	AUC	F1 score
AR-1:						
Original	0.942(0.041)	0.976(0.033)	0.965(0.025)	0.942(0.041)	0.976(0.033)	0.965(0.025)
$\epsilon = 0.1$						
DPFM$_1$	**0.907**(0.048)	0.921(0.068)	**0.947**(0.029)	**0.928**(0.042)	0.959(0.040)	**0.958**(0.026)
DPFM$_2$	0.904(0.048)	0.918(0.075)	0.945(0.029)	**0.928**(0.042)	**0.960**(0.042)	**0.958**(0.026)
DPFM$_3$	0.904(0.048)	0.923(0.067)	0.945(0.029)	**0.928**(0.043)	**0.960**(0.043)	**0.958**(0.026)
DPFM$_4$	0.906(0.049)	**0.926**(0.061)	0.946(0.029)	0.926(0.043)	**0.960**(0.039)	0.956(0.027)
DPRP	0.826(0.054)	0.479(0.106)	0.904(0.032)	0.824(0.099)	0.518(0.211)	0.884(0.115)
PCAPPD	0.831(0.054)	0.452(0.155)	0.906(0.033)	0.828(0.092)	0.471(0.212)	0.897(0.089)
PrivBay	0.836(0.051)	0.513(0.104)	0.910(0.031)	0.831(0.054)	0.524(0.174)	0.907(0.032)
PrivPGM	0.827(0.056)	0.499(0.116)	0.904(0.034)	0.809(0.077)	0.473(0.198)	0.892(0.050)
$\epsilon = 5$						
DPFM$_1$	**0.906**(0.051)	0.923(0.070)	**0.946**(0.031)	0.926(0.043)	0.961(0.039)	**0.957**(0.026)
DPFM$_2$	0.904(0.050)	0.918(0.076)	0.945(0.030)	**0.927**(0.042)	0.959(0.046)	**0.957**(0.025)
DPFM$_3$	0.905(0.051)	0.924(0.070)	0.945(0.031)	0.925(0.044)	0.960(0.040)	0.956(0.026)
DPFM$_4$	**0.906**(0.050)	**0.925**(0.069)	**0.946**(0.030)	**0.927**(0.043)	**0.963**(0.036)	**0.957**(0.026)
DPRP	0.827(0.054)	0.498(0.120)	0.904(0.032)	0.841(0.052)	0.557(0.224)	0.913(0.031)
PCAPPD	0.832(0.053)	0.458(0.152)	0.907(0.032)	0.831(0.076)	0.557(0.194)	0.904(0.060)
PrivBay	0.832(0.049)	0.493(0.125)	0.907(0.029)	0.839(0.052)	0.576(0.190)	0.911(0.031)
PrivPGM	0.835(0.054)	0.512(0.138)	0.909(0.033)	0.838(0.052)	0.432(0.183)	0.911(0.031)
Exch:						
Original	0.940(0.039)	0.977(0.026)	0.964(0.024)	0.940(0.039)	0.977(0.026)	0.964(0.024)
$\epsilon = 0.1$						
DPFM$_1$	0.907(0.047)	0.923(0.068)	0.946(0.028)	0.928(0.039)	0.961(0.039)	0.958(0.024)
DPFM$_2$	0.909(0.0546)	**0.924**(0.066)	**0.948**(0.027)	0.928(0.041)	**0.963**(0.039)	0.958(0.024)
DPFM$_3$	**0.910**(0.045)	**0.924**(0.067)	**0.948**(0.026)	**0.929**(0.040)	0.962(0.039)	0.958(0.024)
DPFM$_4$	0.909(0.045)	**0.924**(0.068)	0.947(0.027)	**0.929**(0.040)	0.962(0.039)	**0.959**(0.024)
DPRP	0.829(0.054)	0.509(0.116)	0.905(0.033)	0.832(0.082)	0.492(0.203)	0.899(0.089)
PCAPPD	0.833(0.054)	0.463(0.155)	0.908(0.032)	0.828(0.091)	0.454(0.205)	0.897(0.088)
PrivBay	0.835(0.050)	0.497(0.120)	0.909(0.030)	0.833(0.052)	0.514(0.193)	0.908(0.031)
PrivPGM	0.832(0.053)	0.487(0.159)	0.907(0.033)	0.805(0.086)	0.479(0.201)	0.887(0.065)
$\epsilon = 5$						
DPFM$_1$	**0.908**(0.046)	**0.925**(0.069)	**0.947**(0.028)	**0.933**(0.039)	0.961(0.041)	**0.961**(0.024)
DPFM$_2$	0.907(0.046)	0.924(0.068)	**0.947**(0.027)	0.931(0.041)	**0.964**(0.037)	0.960(0.025)
DPFM$_3$	0.906(0.048)	0.921(0.069)	0.946(0.029)	0.932(0.039)	0.962(0.039)	0.960(0.023)
DPFM$_4$	0.907(0.048)	0.919(0.070)	0.946(0.028)	**0.933**(0.038)	0.961(0.040)	**0.961**(0.023)
DPRP	0.826(0.053)	0.517(0.101)	0.904(0.032)	0.841(0.050)	0.522(0.216)	0.913(0.030)
PCAPPDP	0.833(0.055)	0.460(0.154)	0.907(0.033)	0.830(0.076)	0.547(0.223)	0.903(0.061)
PrivBay	0.832(0.047)	0.512(0.120)	0.907(0.028)	0.835(0.050)	0.559(0.185)	0.909(0.030)
PrivPGM	0.835(0.047)	0.504(0.128)	0.910(0.028)	0.835(0.048)	0.464(0.181)	0.909(0.029)

Table 4: Average criteria of three classifiers and standard deviation in parentheses for synthetic testing data from eight algorithms and raw validating data in setting (II).

	Testing Data			Validating Data		
	accuracy	AUC	F1 score	accuracy	AUC	F1 score
AR-1:						
Original	0.925(0.042)	0.970(0.028)	0.957(0.025)	0.925(0.042)	0.970(0.028)	0.957(0.025)
$\epsilon = 0.1$						
DPFM_1	**0.873**(0.047)	0.853(0.102)	**0.929**(0.027)	**0.898**(0.046)	0.933(0.058)	**0.942**(0.027)
DPFM_2	0.872(0.049)	0.850(0.104)	**0.929**(0.029)	0.896(0.047)	**0.934**(0.057)	0.941(0.027)
DPFM_3	0.872(0.049)	0.850(0.103)	**0.929**(0.029)	0.896(0.047)	**0.934**(0.057)	0.941(0.027)
DPFM_4	0.872(0.048)	**0.855**(0.100)	**0.929**(0.028)	**0.898**(0.046)	**0.934**(0.057)	**0.942**(0.027)
DPRP	0.843(0.043)	0.518(0.149)	0.914(0.025)	0.856(0.038)	0.510(0.161)	0.922(0.022)
PCAPPD	0.847(0.043)	0.430(0.114)	0.916(0.025)	0.856(0.038)	0.477(0.162)	0.922(0.022)
PrivBay	0.850(0.041)	0.503(0.120)	0.918(0.024)	0.848(0.042)	0.500(0.147)	0.917(0.024)
PrivPGM	0.849(0.039)	0.504(0.135)	0.918(0.023)	0.848(0.042)	0.479(0.142)	0.917(0.026)
$\epsilon = 5$						
DPFM_1	0.870(0.047)	0.850(0.099)	0.928(0.028)	0.895(0.047)	**0.932**(0.063)	0.940(0.028)
DPFM_2	**0.872**(0.049)	**0.856**(0.101)	**0.929**(0.029)	**0.896**(0.048)	0.931(0.0461)	0.941(0.028)
DPFM_3	**0.872**(0.049)	0.852(0.103)	0.924(0.029)	**0.896**(0.048)	0.931(0.064)	**0.941**(0.028)
DPFM_4	0.871(0.046)	0.854(0.100)	0.928(0.027)	**0.896**(0.048)	**0.932**(0.064)	**0.941**(0.028)
DPRP	0.840(0.049)	0.501(0.153)	0.912(0.030)	0.856(0.038)	0.520(0.172)	0.922(0.022)
PCAPPD	0.847(0.042)	0.446(0.111)	0.917(0.025)	0.856(0.038)	0.524(0.179)	0.922(0.022)
PrivBay	0.867(0.094)	0.584(0.218)	0.925(0.051)	0.863(0.088)	0.603(0.155)	0.925(0.051)
PrivPGM	0.851(0.040)	0.486(0.126)	0.919(0.023)	0.850(0.040)	0.496(0.142)	0.919(0.023)
Exch:						
Original	0.943(0.038)	0.977(0.033)	0.966(0.023)	0.943(0.038)	0.977(0.033)	0.966(0.023)
$\epsilon = 0.1$						
DPFM_1	**0.882**(0.051)	0.880(0.088)	**0.934**(0.030)	0.912(0.043)	0.952(0.043)	0.949(0.026)
DPFM_2	0.881(0.049)	**0.883**(0.085)	**0.934**(0.029)	0.913(0.045)	0.953(0.042)	0.949(0.027)
DPFM_3	0.881(0.049)	**0.883**(0.085)	0.933(0.029)	0.912(0.045)	0.953(0.042)	0.949(0.027)
DPFM_4	**0.882**(0.048)	0.877(0.088)	**0.934**(0.028)	**0.916**(0.043)	**0.955**(0.045)	**0.951**(0.026)
DPRP	0.848(0.049)	0.539(0.148)	0.916(0.029)	0.842(0.087)	0.505(0.172)	0.906(0.086)
PCAPPD	0.848(0.049)	0.440(0.119)	0.917(0.029)	0.856(0.044)	0.480(0.169)	0.922(0.027)
PrivBay	0.848(0.049)	0.502(0.162)	0.917(0.029)	0.843(0.055)	0.520(0.169)	0.914(0.034)
PrivPGM	0.847(0.042)	0.506(0.151)	0.917(0.025)	0.849(0.050)	0.466(0.171)	0.917(0.031)
$\epsilon = 5$						
DPFM_1	0.878(0.048)	0.879(0.086)	0.932(0.028)	0.912(0.047)	**0.955**(0.044)	0.949(0.028)
DPFM_2	**0.880**(0.050)	0.882(0.086)	**0.933**(0.029)	**0.914**(0.046)	**0.955**(0.045)	**0.950**(0.027)
DPFM_3	**0.880**(0.049)	**0.883**(0.085)	**0.933**(0.029)	0.913(0.046)	0.954(0.045)	**0.950**(0.027)
DPFM_4	0.879(0.048)	0.882(0.084)	0.932(0.028)	0.911(0.046)	0.952(0.047)	0.948(0.028)
DPRP	0.843(0.049)	0.535(0.161)	0.913(0.029)	0.843(0.087)	0.521(0.190)	0.906(0.086)
PCAPPD	0.847(0.047)	0.448(0.111)	0.917(0.028)	0.856(0.044)	0.517(0.177)	0.922(0.026)
PrivBay	0.850(0.094)	0.610(0.140)	0.917(0.056)	0.846(0.100)	0.538(0.063)	0.915(0.059)
PrivPGM	0.850(0.046)	0.528(0.120)	0.918(0.027)	0.850(0.039)	0.510(0.153)	0.918(0.023)

K. Real data analysis results

The results of $\epsilon = 0.1$ and 5 for the three real data analyses are shown in Tables 5 - 7. These results show that the proposed method can generate synthetic data with high utility for various types of the original data. Specifically, from Table 5, we observe that the maximum discrepancies of the accuracy, AUC and F1 score of the proposed method compared with those of the original data are 0.027, 0.054 and 0.015, respectively. In addition, the maximum discrepancies of the three criteria from other methods compared with those of the original data are 0.289, 0.368 and 0.080, respectively. From Table 7, we observe that the maximum discrepancies of the accuracy, AUC and F1 score of the proposed method compared with those of the original data are 0.120, 0.157 and 0.157, respectively. In addition, the maximum discrepancies of the three criteria from other methods compared with those of the original data are 0.263, 0.368 and 0.334, respectively. This indicates that although the proposed method has utility loss in order to satisfy a certain privacy protection requirement, the utility loss is lower than that of the competing methods. Tables 6 shows similar performance results.

Table 5: Average criteria of three classifiers and standard deviation in parentheses for testing synthetic data from five algorithms and validating data from the Census Income Data.

	Testing Data			Validating Data		
	accuracy	AUC	F1 score	accuracy	AUC	F1 score
Original	0.812(0.027)	0.848(0.035)	0.878(0.018)	0.812(0.027)	0.848(0.035)	0.878(0.018)
			$\epsilon = 0.1$			
DPFM	**0.785**(0.031)	**0.795**(0.040)	**0.864**(0.021)	**0.802**(0.031)	**0.829**(0.035)	**0.874**(0.021)
DPRP	0.713(0.029)	0.513(0.046)	0.828(0.021)	0.523(0.100)	0.523(0.070)	0.798(0.172)
PCAPPD	0.708(0.021)	0.486(0.046)	0.825(0.015)	0.725(0.072)	0.529(0.120)	0.823(0.102)
PrivBay	0.722(0.040)	0.494(0.053)	0.834(0.028)	0.716(0.042)	0.492(0.100)	0.832(0.029)
PrivPGM	0.719(0.039)	0.493(0.070)	0.833(0.027)	0.710(0.051)	0.495(0.112)	0.825(0.038)
			$\epsilon = 5$			
DPFM	**0.785**(0.031)	**0.794**(0.042)	**0.863**(0.022)	**0.802**(0.030)	**0.828**(0.037)	**0.874**(0.021)
DPRP	0.716(0.029)	0.501(0.052)	0.830(0.020)	0.722(0.070)	0.489(0.135)	0.826(0.084)
PCAPPD	0.707(0.022)	0.494(0.043)	0.825(0.015)	0.733(0.041)	0.487(0.115)	0.824(0.032)
PrivBay	0.728(0.037)	0.508(0.066)	0.839(0.025)	0.713(0.039)	0.480(0.096)	0.829(0.027)
PrivPGM	0.709(0.033)	0.486(0.075)	0.824(0.023)	0.699(0.051)	0.490(0.097)	0.816(0.037)

Table 6: Average criteria of three classifiers and standard deviation in parentheses for testing synthetic data from five algorithms and validating data from the Absenteeism at Work Data.

	Testing Data			Validating Data		
	accuracy	AUC	F1 score	accuracy	AUC	F1 score
Original	0.752(0.050)	0.831(0.049)	0.730(0.063)	0.766(0.047)	0.840(0.044)	0.733(0.061)
$\epsilon = 0.1$						
DPFM	**0.632**(0.040)	**0.675**(0.046)	**0.573**(0.055)	**0.655**(0.040)	**0.717**(0.044)	**0.596**(0.055)
DPRP	0.501(0.041)	0.498(0.050)	0.418(0.050)	0.502(0.045)	0.496(0.064)	0.253(0.314)
PCAPPD	0.506(0.036)	0.494(0.045)	0.402(0.052)	0.506(0.046)	0.498(0.059)	0.166(0.280)
PrivBay	0.511(0.068)	0.497(0.086)	0.350(0.104)	0.508(0.066)	0.490(0.077)	0.339(0.135)
PrivPGM	0.498(0.053)	0.492(0.061)	0.276(0.044)	0.503(0.067)	0.490(0.108)	0.298(0.154)
$\epsilon = 5$						
DPFM	**0.632**(0.040)	**0.674**(0.045)	**0.572**(0.055)	**0.656**(0.041)	**0.717**(0.044)	**0.597**(0.056)
DPRP	0.506(0.043)	0.497(0.050)	0.408(0.061)	0.516(0.046)	0.510(0.068)	0.200(0.296)
PCAPPD	0.505(0.036)	0.494(0.043)	0.396(0.051)	0.512(0.045)	0.512(0.068)	0.199(0.258)
PrivBay	0.508(0.047)	0.489(0.072)	0.358(0.092)	0.515(0.073)	0.512(0.092)	0.357(0.133)
PrivPGM	0.497(0.053)	0.483(0.055)	0.292(0.080)	0.527(0.061)	0.515(0.081)	0.349(0.084)

Table 7: Average criteria of three classifiers and standard deviation in parentheses for testing synthetic data from five algorithms and validating data from the Breast Cancer Data.

	Testing Data			Validating Data		
	accuracy	AUC	F1 score	accuracy	AUC	F1 score
Original	0.733(0.074)	0.803(0.080)	0.752(0.073)	0.733(0.074)	0.803(0.080)	0.752(0.073)
$\epsilon = 0.1$						
DPFM	**0.678**(0.093)	**0.735**(0.105)	**0.707**(0.085)	**0.669**(0.103)	**0.737**(0.115)	**0.711**(0.094)
DPRP	0.494(0.100)	0.478(0.131)	0.577(0.097)	0.511(0.085)	0.505(0.148)	0.533(0.303)
PCAPPD	0.503(0.097)	0.483(0.113)	0.563(0.112)	0.519(0.090)	0.523(0.165)	0.418(0.341)
PrivBay	0.527(0.133)	0.531(0.161)	0.588(0.150)	0.508(0.151)	0.511(0.207)	0.553(0.210)
PrivPGM	0.502(0.180)	0.484(0.205)	0.566(0.193)	0.499(0.125)	0.463(0.214)	0.561(0.170)
$\epsilon = 5$						
DPFM	**0.681**(0.093)	**0.737**(0.107)	**0.710**(0.084)	**0.673**(0.099)	**0.740**(0.114)	**0.714**(0.090)
DPRP	0.495(0.107)	0.487(0.143)	0.578(0.108)	0.531(0.079)	0.505(0.144)	0.538(0.302)
PCAPPD	0.509(0.105)	0.494(0.129)	0.568(0.116)	0.539(0.095)	0.531(0.138)	0.622(0.111)
PrivBay	0.535(0.132)	0.517(0.180)	0.604(0.159)	0.514(0.117)	0.499(0.154)	0.514(0.227)
PrivPGM	0.470(0.155)	0.435(0.205)	0.520(0.218)	0.511(0.124)	0.520(0.190)	0.602(0.158)

References

Nazmiye Ceren Abay, Yan Zhou, Murat Kantarcioglu, Bhavani Thuraisingham, and Latanya Sweeney. Privacy preserving synthetic data release using deep learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 510–526, 2018.

Gergely Acs, Luca Melis, Claude Castelluccia, and Emiliano De Cristofaro. Differentially private mixture of generative neural networks. *IEEE Transactions on Knowledge and Data Engineering*, 31(6):1109–1121, 2019.

Seung C. Ahn and Alex R. Horenstein. Eigenvalue ratio test for the number of factors. *Econometrica*, 81(3):1203–1227, 2013.

Héber H. Arcolezi. Production of categorical data verifying differential privacy: Conception and applications to machine learning. *arXiv preprint arXiv:2204.00850*, 2022.

Raman Arora, Vladimir braverman, and Jalaj Upadhyay. Differentially private robust low-rank approximation. In *Advances in Neural Information Processing Systems*, volume 31, pages 1–9, 2018.

Marco Avella-Medina. Privacy-preserving parametric inference: a case for robust statistics. *Journal of the American Statistical Association*, 116(534):969–983, 2021.

Jordan Awan and Aleksandra Slavković. Structure and sensitivity in differential privacy: Comparing k-norm mechanisms. *Journal of the American Statistical Association*, 116 (534):935–954, 2021.

Jushan Bai. Inferential theory for factor models of large dimensions. *Econometrica*, 71(1): 135–171, 2003.

Jushan Bai and Serena Ng. Determining the number of factors in approximate factor models. *Econometrica*, 70(1):191–221, 2002.

Ergute Bao, Xiaokui Xiao, Jun Zhao, Dongping Zhang, and Bolin Ding. Synthetic data generation with differential privacy via bayesian networks. *Journal of Privacy and Confidentiality*, 11(3):1–20, 2021.

Xuan Bi and Xiaotong Shen. Distribution-invariant differential privacy. *Journal of Econometrics*, 2022. https://doi.org/10.1016/j.jeconom.2022.05.004.

Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. The Johnson-Lindenstrauss transform itself preserves differential privacy. In *the 2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 410–419, 2012.

Zhiqi Bu, Jinshuo Dong, Qi Long, and Weijie Su. Deep learning with Gaussian differential privacy. *Harvard Data Science Review*, 2(3), 2020. https://doi.org/10.1162/99608f92.cfc5dd25.

T Tony Cai, Yichen Wang, and Linjun Zhang. The cost of privacy: Optimal rates of convergence for parameter estimation with differential privacy. *The Annals of Statistics*, 49(5):2825–2850, 2021.

Thee Chanyaswad, Changchang Liu, and Prateek Mittal. RON-Gauss: Enhancing utility in non-interactive private data release. In *Proceedings of Privacy Enhancing Technologies*, volume 1, pages 26–46, 2019.

Dingfan Chen, Tribhuvanesh Orekondy, and Mario Fritz. GS-WGAN: A gradient-sanitized approach for learning differentially private generators. *Advances in Neural Information Processing Systems*, 33:12673–12684, 2020.

Rui Chen, Qian Xiao, Yu Zhang, and Jianliang Xu. Differentially private high-dimensional data publication via sampling-based inference. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 129–138, 2015.

Graham Cormode, Samuel Maddock, and Carsten Maple. Frequency estimation under local differential privacy. *Proceedings of the VLDB Endowment*, 14(11):2046–2058, 2021.

Jinshuo Dong, Aaron Roth, and Weijie J Su. Gaussian differential privacy. *arXiv preprint arXiv:1905.02383*, 2019.

John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 429–438, 2013.

John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. Minimax optimal procedures for locally private estimation. *Journal of the American Statistical Association*, 113(521): 182–201, 2018.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Cynthia Dwork and Adam Smith. Differential privacy for statistics: What we know and what we want to learn. *Journal of Privacy and Confidentiality*, 1(2):135–154, 2010.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, pages 265–284, 2006.

Cynthia Dwork, Kunal Talwar, Abhradeep Thakurta, and Li Zhang. Analyze Gauss: Optimal bounds for privacy-preserving principal component analysis. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, page 11–20, 2014.

Alexandre Evfimievski, Johannes Gehrke, and Ramakrishnan Srikant. Limiting privacy breaches in privacy preserving data mining. In *Proceedings of the Twenty-Second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 211–222, 2003.

Lorenzo Frigerio, Anderson Santana de Oliveira, Laurent Gomez, and Patrick Duverger. Differentially private generative adversarial networks for time series, continuous, and discrete open data. In *IFIP International Conference on ICT Systems Security and Privacy Protection*, pages 151–164, 2019.

Harvey Goldstein and William Browne. Multilevel factor analysis models for continuous and discrete data. In *Contemporary Psychometrics: A Festschrift for Roderick P McDonald*, pages 453 – 475, 2005.

Lovedeep Gondara and Ke Wang. Differentially private small dataset release using random projections. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence*, volume 124, pages 639–648, 2020.

Robert Hall, Larry Wasserman, and Alessandro Rinaldo. Random differential privacy. *Journal of Privacy and Confidentiality*, 4(2), 2013. https://doi.org/10.29012/jpc.v4i2.621.

Moritz Hardt and Guy N Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 61–70, 2010.

Hafiz Imtiaz and Anand D. Sarwate. Distributed differentially private algorithms for matrix and tensor factorization. *IEEE Journal of Selected Topics in Signal Processing*, 12(6): 1449–1464, 2018.

Wuxuan Jiang, Cong Xie, and Zhihua Zhang. Wishart mechanism for differentially private principal components analysis. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, page 1730–1736, 2016.

Xiaoqian Jiang, Zhanglong Ji, Shuang Wang, Noman Mohammed, Samuel Cheng, and Lucila Ohno-Machado. Differential-private data publishing through component analysis. *Transactions on Data Privacy*, 6(1):19–34, 2013.

Haoran Li, Li Xiong, and Xiaoqian Jiang. Differentially private synthesization of multi-dimensional data using copula functions. *Advances in database technology : proceedings. International Conference on Extending Database Technology*, 2014:475–486, 2014.

Chong Liu, Yuqing Zhu, Kamalika Chaudhuri, and Yu-Xiang Wang. Revisiting model-agnostic private learning: Faster rates and active learning. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, pages 838–846, 2021.

Ryan Mckenna, Daniel Sheldon, and Gerome Miklau. Graphical-model based estimation and inference for differential privacy. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4435–4444, 2019.

Ryan McKenna, Gerome Miklau, and Daniel Sheldon. Winning the NIST contest: A scalable and general approach to differentially private synthetic data. *Journal of Privacy and Confidentiality*, 11(3):1–29, 2021.

Frank Mcsherry and Anna Karlin. *Spectral Methods for Data Analysis.* PhD thesis, USA, 2004.

Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science*, pages 94–103, 2007.

Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*, pages 263–275, 2017.

Alexei Onatski. Determining the number of factors from empirical distribution of eigenvalues. *The Review of Economics and Statistics*, 92(4):1004–1016, 2010.

Haoyue Ping, Julia Stoyanovich, and Bill Howe. Datasynthesizer: Privacy-preserving synthetic datasets. In *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*, page 1–5, 2017.

Angelika Rohde and Lukas Steinberger. Geometrizing rates of convergence under local differential privacy constraints. *The Annals of Statistics*, 48(5):2646–2670, 2020.

Xiaotong Shen, Xuan Bi, and Rex Shen. Data Flush. *Harvard Data Science Review*, 4(2), 2022. https://hdsr.mitpress.mit.edu/pub/x1ozqj10.

Anders Skrondal and Sophia Rabe-Hesketh. *Generalized Latent Variable Modeling: Multilevel, Longitudinal, and Structural Equation Models.* Chapman and Hall/CRC, 2004.

Xin-Yuan Song, Zhao-Hua Lu, Jing-Heng Cai, and Edward Hak-Sing Ip. A Bayesian Modeling Approach for Generalized Semiparametric Structural Equation Models. *Psychometrika*, 78(4):624–647, 2013.

Uthaipon Tao Tantipongpipat, Chris Waites, Digvijay Boob, Amaresh Ankit Siva, and Rachel Cummings. Differentially private synthetic mixed-type data generation for unsupervised learning. In *2021 12th International Conference on Information, Intelligence, Systems and Applications*, pages 1–9, 2021.

Terence Tao. *Topics in Random Matrix Theory*, volume 132 of *Graduate studies in mathematics.* American Mathematical Society, 2012.

Jalaj Upadhyay. The price of privacy for low-rank factorization. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, page 4180–4191, 2018.

Tianhao Wang, Jeremiah Blocki, Ninghui Li, and Somesh Jha. Locally differentially private protocols for frequency estimation. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 729–745, 2017.

Jianhong Wu. Robust determination for the number of common factors in the approximate factor models. *Economics Letters*, 144:102–106, 2016.

Qiang Xia, Rubing Liang, and Jianhong Wu. Transformed contribution ratio test for the number of factors in static approximate factor models. *Computational Statistics and Data Analysis*, 112:235–241, 2017.

Xiaokui Xiao, Guozhang Wang, and Johannes Gehrke. Differential privacy via wavelet transforms. *IEEE Transactions on Knowledge and Data Engineering*, 23(8):1200–1214, 2011.

Chugui Xu, Ju Ren, Yaoxue Zhang, Zhan Qin, and Kui Ren. Dppro: Differentially private high-dimensional data release via random projection. *IEEE Transactions on Information Forensics and Security*, 12(12):3081–3093, 2017.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *International Conference on Learning Representations*, pages 1–21, 2019.

Jun Zhang, Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Privbayes: Private data release via bayesian networks. *ACM Transactions on Database Systems*, 42(4):1–41, 2017.

Qinqing Zheng, Shuxiao Chen, Qi Long, and Weijie Su. Federated f-differential privacy. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, pages 2251–2259, 2021.

Shuheng Zhou, Katrina Ligett, and Larry Wasserman. Differential privacy with compression. In *2009 IEEE International Symposium on Information Theory*, pages 2718–2722, 2009.