High-dimensional Dataset Publishing Methods Satisfying Personalized Differential Privacy Based on Bayesian Network

Churou Deng | dengchurou@gmail.com

June 23, 2025

High-dimensional Dataset Publishing Method Satisfying Personalized Differential Privacy Based on Bayesian Network

Abstract: In this paper, we study the problem of publishing datasets that satisfy differential privacy. In this problem, we wish to publish a dataset and ensure that the publication algorithm meets the ϵ-differential privacy protection requirements. Most existing studies based on this problem directly set the privacy requirements of each piece of data and each attribute in a dataset as the same. However, in real life, the sensitivity of individuals and attributes in a dataset could be different, which may lead to various privacy requirements for different tuples and various privacy requirements for different attributes. Therefore, this paper proposes a hierarchical personalized differential privacy algorithm (HPPrivBayes) based on the classical privacy-preserving data publishing method PrivBayes, so that the new algorithm could publish a dataset satisfying differential privacy and also meet personalized privacy needs. This approach quantifies the privacy requirements of each attribute and the importance of different values in some attributes, then allocates the corresponding privacy-preserving budgets for different levels of requirements. In addition, since the PrivBayes method has defects in constructing Bayesian networks, to improve the quality of the obtained Bayesian networks, HPPrivBayes will optimize the Bayesian network construction process of PrivBayes based on the average mutual information. The experimental results show that compared to PrivBayes, our method not only improves the quality of Bayesian network construction but also guarantees personalized privacy budget allocation, ensuring the availability and accuracy of the generated dataset.

Keywords: Differential Privacy, Personalized Privacy Budget Allocation, Bayesian Network, Mutual Information, Data Publishing

基于贝叶斯网络的满足差分隐私的高维数据发布方法

摘要: 本文研究了如何满足差分隐私的数据集发布问题。在这个问题中，我们希望发布一个数据集，并确保发布算法满足 ϵ-差分隐私保护要求。基于这个问题的现有研究大多直接将数据集中每条数据和每个属性的隐私要求设置为相同。然而，在现实生活中，数据集中的个体或属性的敏感度可能不同，这可能导致不同数据元组的隐私要求不同，不同属性的隐私要求也不同。因此，本文在经典隐私保护数据发布方法 PrivBayes 的基础上，提出了一种分级的个性化差分隐私算法（HPPrivBayes），使新算法既能发布满足差分隐私的数据集，又能满足各个体与属性的个性化的隐私需求。这种方法可以量化每个属性的隐私要求以及部分属性中不同值的重要性，并针对不同层次的需求分配相应的隐私保护预算。此外，由于 PrivBayes 方法本身在构建贝叶斯网络时存在缺陷，为了提高得到的贝叶斯网络的质量，HPPrivBayes 将根据平均互信息对 PrivBayes 的贝叶斯网络构建过程进行优化。实验结果表明，相较于 PrivBayes，我们的方法不仅提升了贝叶斯网络的构建质量，同时保证了个性化的隐私预算分配，确保了生成数据集的可用性和准确性。

关键字: 差分隐私，个性化隐私预算分配，贝叶斯网络，互信息，数据发布

Contents

1. Introduction

1.1. Research Background and Significance

In the era of big data, people's demand and ability to generate, collect, and use data are constantly increasing. Consequently, data privacy issues have attracted wide attention. In reality, people's research demand for data leads to the frequent flow and dissemination of data, which often contains a large amount of personal sensitive information (such as medical data, financial investment data, etc.). Therefore, publishing unprocessed datasets may lead to serious information disclosure risks to the public and illegal risks for data publishers. In recent years, many countries have introduced legal provisions to protect privacy security, for example, China has successively formulated and implemented laws and regulations such as the Cybersecurity Law and the Personal Information Protection Law. All these constrain and standardize the data publishing behavior (Tang, 2019a). If the privacy leakage problem of data publishing cannot be addressed, the publishing of data will be restricted, thus hindering the development of data research and its application.

1.1.1. Differential Privacy

The data publishing method of privacy protection provides a feasible solution to address the problem of personal privacy leakage caused by data publishing. Differential privacy is a concept that has been widely used in privacy-preserving data publishing (PPDP) since it was first proposed in 2006 (Dwork, 2006), which can quantitatively analyze privacy and better protect the usability of data while reducing the risk of data leakage (Zhang and Chen, 2021). Compared with the traditional data anonymization protection methods (such as K-anonymity (Sweeney, 2002), L-diversity (Machanavajjhala et al., 2007), etc.), which cannot resist attacks under all background knowledge, differential privacy has attracted more attention because of its strict mathematical definition and logical proof.

1.1.2. High-Dimensional Dataset Publishing based on Bayesian Network

Because of the large data volumes and high dimensions of data nowadays, the curse of dimensionality usually causes the ineffectiveness of traditional methods in the publishing and analysis of big data. To address this problem, the idea of data dimensionality reduction is often used, which is to map high-dimensional data to low-dimensional space and perform data analysis in low-dimensional space. In 2000, Mu (Mu et al., 2000) first proposed the application of the Bayesian network to data publishing. A Bayesian network is a directed acyclic graph (DAG), which is composed of nodes (representing variables) and edges (representing dependencies) connecting nodes, and can effectively handle dependencies between variables. Because the Bayesian network can more easily ensure the consistency and completeness of data attributes, it has a wider application space in high-dimensional data compared with other dimension reduction methods. At the same time, the Bayesian network uses the mutual information size between

attribute nodes to represent the dependence between attributes, which combines prior knowledge and sample knowledge and is more suitable for sparsely high-dimensional data.

1.2. Related Work

1.2.1. PrivBayes

Since the concept of differential privacy was first proposed, researchers at home and abroad have made certain achievements in the field of high-dimensional data publishing satisfying differential privacy. In 2017, Zhang et al. proposed a Bayesian network-based differential privacy protection publishing method PrivBayes (Zhang et al., 2017), which soon attracted the interest of other researchers. PrivBayes method first uses the constructed Bayesian network to reduce the dimension of high-dimensional data and then the exponential and Laplacian mechanisms to add noise to the constructed Bayesian network to achieve differential privacy protection.

However, the Bayesian network constructed by this method is not unique and accurate enough because the first node of the network is randomly selected when constructed. Suppose the attackers access the published synthetic dataset multiple times, in that case, the attacker can infer the information of the original dataset based on the different synthetic datasets generated by the algorithm from the multiple accesses, which may lead to data leakage. Besides, this method directly allocates the privacy budget without considering the different privacy needs of different individuals and attributes which may thus lead to unreasonable addition of noise, which harms the usability and privacy of data. Subsequently, many scholars have made improvements based on this method.

1.2.2. Improvements based on PrivBayes

Aiming at the problem of random selection of the first node, Wang et al. (Wang et al., 2016) proposed a weighted Bayesian network-based privacy data publishing method in 2016, which adds a weight value to each attribute field node according to their importance in the published dataset. Combining the K2 scoring function and mutual information, the Bayesian networks constructed become more accurate. This method further optimizes the order of adding noise to attribute field nodes, which can significantly improve the accuracy of the synthetic datasets and algorithm efficiency compared with the PrivBayes.

Tang Shiyi (Tang, 2018) proposed to apply wavelet transform to data before adding noise and replace mutual information with an improved F function, to obtain a more accurate dependency relationship, and thus improve the accuracy of the constructed Bayesian networks. The core idea of the algorithm is to use the low-dimensional marginal distribution to describe the Bayesian network, add the same Laplacian noise to each item of the low-dimensional marginal distribution, and then sample to form a synthetic dataset with noise.

Aiming at the problem that unreasonable noise is added to the PriveBayes algorithm, which

affects the usability of data, Li et al (Li and Ma, 2018) proposed the SS-PrivBayes (smooth sensitivity privacy Bayes) algorithm. When analyzing the actual dataset, local dataset changes are also considered to obtain different attribute sensitivities, which is beneficial to reducing noise intake during the differential privacy process and improving the usability of data release. Besides, aiming at the low-efficiency problem of the PrivBayes algorithm in constructing Bayesian network search space, an efficient Bayesian network search space algorithm PBCPC (privacy Bayesian candidate parents and children) is proposed. This algorithm obtains the parent-child set of target variables by a heuristic method. When the number of attributes is large, the algorithm's efficiency is higher than that of PriveBayes.

Zhang et al. (Zhang et al., 2019) proposed a method of differential privacy publishing for high-dimensional data (PrivMN) based on Markov networks to address the problem of data-sparse in differential privacy publishing for high-dimensional data. In this algorithm, the Markov model measures the dependence between attributes, and the distribution of differential privacy of high-dimensional data is calculated by combining the graph approximate inference algorithm. The algorithm can effectively reduce the noise intake and address the problem of high computational complexity in precise reasoning.

Given the situation that the Bayesian network is not unique and the privacy budget is not allocated reasonably in the existing differential privacy model based on the construction of Bayesian networks, Chen (Chen, 2021) proposed a personalized differential privacy algorithm CSAPrivBayes based on improved Bayesian network. This algorithm allocates the privacy budget by dividing attributes into two parts according to their mutual information to an associated sensitive attribute, and it improves the Bayesian network's initial node random selection mechanism. However, this algorithm simply classifies the privacy requirements of attributes into two hierarchies and fails to consider that different values in each attribute may also have different privacy requirements.

In 2023, Lu proposed a differential privacy data publishing algorithm ELPrivBayes based on the Bayesian network to further reduce the computational overhead and improve the quality of constructed Bayesian networks. This article reduces the time complexity by constructing a correlation matrix that stores mutual information between attributes to avoid redundant computation of mutual information during the iteration process of the structure learning algorithm (Lu et al., 2023). In addition, this method optimizes the order of nodes entering the Bayesian network based on the average mutual information, which in turn reduces the marginal distance of the generated dataset to the original dataset. The article compares the Bayesian models built by ELPrivBayes and Greedy algorithm, which applies the maximum average mutual information (AMI) to select the first node, and finds that the quality of the Bayesian models constructed by

ELPrivBayes is higher than that of the latter, which suggests that the advantage of ELPrivBayes does not come from the optimization of the first node, this advantage is reflected in the optimization of all node selections.

Improvements to PrivBayes given by the above works can be broadly categorized into two groups: 1. improvements targeting the node selection methodology during the construction of Bayesian networks and 2. improvements from the perspective of privacy budget allocation. Both of these perspectives can improve the usability of the synthetic datasets. The first perspective is achieved by optimizing the structure of the constructed Bayesian network, while the second perspective is achieved by more rational addition of noise. However, only a few researchers take both optimization perspectives into account as we mentioned, while none of them consider the different privacy requirements that may exist for different values in each attribute.

1.3. Our contributions

This paper proposes a hierarchical personalized differential privacy algorithm (HPPrivBayes) based on PrivBayes, and we will take both optimization perspectives we mentioned above into account. First, we wish to improve the quality of the obtained networks by optimizing the order in which the different attributes enter the network during the construction of the Bayesian network of PrivBayes. Then, we will grade the privacy requirements twice for all attributes as well as for different values in some of the attributes, respectively, and then add different levels of noise to meet the individualized differential privacy according to the different levels of privacy-preserving requirements.

2. Preliminaries

This section reviews three concepts closely related to our work, namely, differential privacy, Bayesian networks, and PrivBayes.

Definition 1 (ϵ-Differential Privacy (Dwork et al., 2006)). Let ϵ be a positive real number, f be a randomized algorithm that satisfies ϵ-differential privacy. For any two datasets D_1 and D_2 that differ only in one tuple, and for any possible output S of f, we have

$$Pr\left[f(D_1) \in S\right] \leq e^{\epsilon} Pr\left[f(D_2) \in S\right], \tag{1}$$

where Pr[$\cdot$] denotes the probability of an event. In the remainder of this paper, we say that two datasets are neighboring if they are different in only one tuple, i.e., the values of one tuple are changed while the rest are identical.

Differential privacy can be realized by two typical mechanisms, the Laplace mechanism and the exponential mechanism. The implementation of these two mechanisms is based on the concept of sensitivity. Therefore, before introducing these two mechanisms, we first introduce the concept of function sensitivity.

Definition 2 (Sensitivity (Dwork et al., 2006)). Let f be a function that maps a dataset into a fixed-size vector of real numbers. The sensitivity of f is defined as

$$S(f) = max \parallel f(D_1) - f(D_2) \parallel_1, \tag{2}$$

where $\parallel \cdot \parallel_1$ denotes the L_1 norm, and D_1, D_2 are any two neighbouring datasets. Intuitively, $S(f)$ measures the maximum possible change in f's output when we modify one arbitrary tuple in f's input.

When f's output is numeric, the Laplace mechanism is usually applied to achieve differential privacy. On the other hand, when f's output is categorical instead of numeric, the Laplace mechanism does not apply, but the exponential mechanism can be used instead.

Definition 3 (Laplace Mechanism (Dwork et al., 2006)). Let a fixed positive real number ϵ be the privacy budget of a given dataset D. Let $S(f)$ be the sensitivity of function f. When the output of f satisfies:

$$A(D) = f(D) + Lap(0, \frac{S(f)}{\epsilon}), \tag{3}$$

then we say that the algorithm A satisfies ϵ-differential privacy, where $Lap(0, \frac{S(f)}{\epsilon})$ is randomized noise satisfying Laplace distribution.

Definition 4 (Laplace Distribution). We say that $x \sim Laplace(\mu, \lambda)$ if x has the density

$$f(x) = \frac{1}{2\lambda} e^{-\frac{|x-\mu|}{\lambda}}, \tag{4}$$

where λ and μ are constant, and $\lambda > 0$. Note that

$$E(x) = \frac{1}{2\lambda} \int_{-\infty}^{\infty} x e^{-\frac{|x-\mu|}{\lambda}} dx = \frac{1}{2} \int_{-\infty}^{\infty} (\lambda y + \mu) e^{-|y|} dy = \mu,$$

$$Var(x) = E(x - E(x))^2 = \frac{1}{2\lambda} \int_{-\infty}^{\infty} (x-\mu)^2 e^{-\frac{|x-\mu|}{\lambda}} dx = \frac{\lambda^2}{2} \int_{-\infty}^{\infty} y^2 e^{-|y|} dy = 2\lambda.$$

Definition 5 (Exponential Mechanism (McSherry and Talwar, 2007)). For a given dataset D, a scoring function $u(D, R)$, if the output of a randomized algorithm M is $r \in R$ with a probability that is proportional to $e^{\frac{\epsilon u(D,R)}{2\Delta u(D,R)}}$, where $\Delta u(D, R)$ is the sensitivity of $u(D, R)$, then we say that algorithm M satisfies ϵ-differential privacy.

Definition 6 (Sequential Composition (McSherry, 2009)). For algorithms M_i each provide ϵ_i-differential privacy, the sequence of $M_i(X)$ provides $(\sum_i \epsilon_i)$-differential privacy.

In other words, any sequence of computations that each provides differential privacy in isolation also provides differential privacy in sequence (Mohammed et al., 2015). Importantly, this is true not only when they are run independently, but even when subsequent computations can incorporate the outcomes of the preceding computations.

Definition 7 (Parallel Composition (McSherry, 2009)). Let M_i each provide ϵ-differential privacy. Let D_i be arbitrary disjoint subsets of the input domain D. The sequence of $M_i(X \cap D_i)$ provides ϵ-differential privacy.

In other words, if the domain of input records is partitioned into disjoint sets, independent of the actual data, and the restrictions of the input data to each part are subjected to differentially private analysis, the ultimate privacy guarantee depends only on the worst of the guarantees of each analysis, not the sum.

Definition 8 (Mutual Information). For any two discrete random variables X and Y, their mutual information can be defined as

$$I(X, Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \cdot log \frac{p(x, y)}{p(x)p(y)},$$

$p(x, y)$ is a the joint probability distribution function of random variables x and y, $p(x)$ and $p(y)$ are marginal probability distribution functions of x and y, respectively.

Definition 9 (Information Entropy). The information entropy of a random variable X over its domain $dom(X)$ is denoted by

$$H(X) = -\sum_{x \in dom(X)} P[X = x] log_2(P[X = x]).$$

2.1. Bayesian Network

As we mentioned in 1.1.1, a Bayesian network is a directed acyclic graph, which is composed of nodes and edges connecting nodes, and can effectively handle dependencies between attributes in a dataset.

A simple way to build a Bayesian network is as follows: first, select a random node from the node set as the starting point of the Bayesian network, and then use the greedy algorithm to select the child-parent pair with the maximum mutual information from the remaining nodes and add it to the Bayesian network. When all nodes are added to the Bayesian network, we get the constructed Bayesian network.

For a Bayesian network on a dataset, nodes represent attributes and edges represent conditional independence among attributes. Let $\mathcal{A} = \{X_1, X_2, ..., X_d\}$ be the set of attributes in a dataset D, and d be the size of $\mathcal{A}$. For a Bayesian network $\mathcal{N}$ on $\mathcal{A}$, ideally, $\mathcal{N}$ should provide an accurate approximation of the tuple distribution in D, i.e., $P_{\mathcal{N}}[\mathcal{A}]$ should be close to $P[\mathcal{A}]$. Thus, to compute $P_{\mathcal{N}}[\mathcal{A}]$, we use conditional distributions $P[X_1|\Pi_1]$, $P[X_2|\Pi_2]$, ..., $P[X_d|\Pi_d]$ to approximate $P[\mathcal{A}]$. Given the parent sets $\Pi_i \subseteq \mathcal{A} \setminus \{X_i\}$ of X_i, if we assume that any $X_k \in \Pi_i$ and $X_j \notin \Pi_i$ are conditionally independent in D, we can get:

$$\begin{aligned} P_{\mathcal{N}}[\mathcal{A}] &= P[X_1, X_2, X_3, ...X_d] \\ &= P[X_1] \cdot P[X_2|X_1] \cdot P[X_3|X_1, X_2] \cdot ... \cdot P[X_d|X_1, X_2, ..., X_{d-1}] \\ &= \prod_{i=1}^{d} P[X_d|\Pi_d] \end{aligned} \tag{5}$$

We define the degree of $\mathcal{N}$ as the maximum size of any parent set Π_i in $\mathcal{N}$. Note that $P_{\mathcal{N}}[\mathcal{A}] = \prod_{i=1}^{d} P[X_i|\Pi_i]$ is approximate to the distribution of dataset $P[\mathcal{A}]$ when the degree of the Bayesian Network is small. If the degree of the Bayesian Network $\mathcal{N}$ is small, then we only need to calculate d low-dimensional distributions $P[X_1]$, $P[X_2|X_1]$, $P[X_3|X_1, X_2]$, ..., $P[X_d|X_1, X_2, ..., X_{d-1}]$ to compute $P_{\mathcal{N}}[\mathcal{A}]$. This property makes low-degree Bayesian networks a core of high-dimensional data publishing. The following is a simple example of a Bayesian network.

Example 1 (Mu et al., 2000). An enterprise needs to evaluate the effectiveness of a new technology in the production process, and the following attributes are available:

1. Age of the operator A
2. Sex of the operator B
3. Effective value of the new technology C
4. Number of unqualified products caused by technical reasons D
5. Number of unqualified products caused by non-technical reasons E

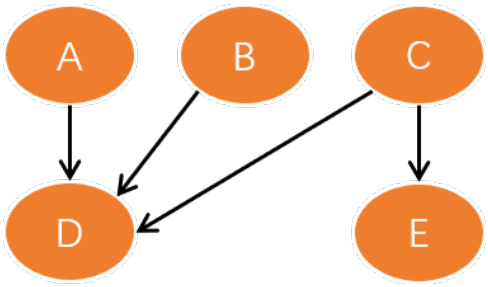


Figure 1: An Example of Bayesian Network on dataset G

In this example, the edge from A to D indicates that the number of unqualified products caused by technical reasons depends on the age of the operator (and also on the sex of the operator and the effective value of the new technology).

2.2. PrivBayes

As mentioned in 1.2.1, PrivBayes is a solution for releasing a high-dimensional dataset D in an ϵ-differential private manner. PrivBayes runs in three steps:

1. Construct a k-degree Bayesian network $\mathcal{N}$ over the attributes in D, using an $\epsilon/2$-differential private algorithm (see algorithm 1). Here, k is a small value that can be chosen automatically by PrivBayes.

Since the simple way of constructing the Bayesian network mentioned in 2.1 does not meet differential privacy, to make this step differentially private, PrivBayes uses an exponential mechanism with the mutual information function I as the score function to select attribute-parent (AP) pair (X_i, Π) from Ω in a private manner. Specifically, PrivBayes first inspects each AP pair $(X_i, \Pi) \in \Omega$, and calculates the mutual information $I(X_i, \Pi)$ between X_i and Π; After that, it samples an AP pair from Ω with a sampling probability of any pair (X_i, Π) proportional to $exp(I(X_i, \Pi)/2\Delta)$, where Δ is a scaling factor and $\Delta = 2(d-1)S(I)/\epsilon$, $S(I)$ denotes the sensitivity of the mutual information function I. This ensures that each invocation of the exponential mechanism satisfies $\epsilon/(2(d-1))$-differential privacy. Given the composability

properties of differential privacy mentioned in Definitions 6 and 7, and the fact that we only invoke the exponential mechanism $(d-1)$ times during the construction of $\mathcal{N}$, it can be verified that the overall process of constructing $\mathcal{N}$ is $\epsilon/2$-differentially private (Zhang et al., 2017).

Algorithm 1 Construct Bayesian network

Input: D, k,
Output: $\mathcal{N}$

Initialize $\mathcal{N} = \emptyset$ and $V = \emptyset$
Randomly select an attribute X_1 from $\mathcal{A}$; add $(X_1, \emptyset)$ to $\mathcal{N}$; add X_1 to V
for $i = 2 \rightarrow d$ **do**
    Initialize $\Omega = \emptyset$
    for each $X_i \in \mathcal{A} \setminus V$ and each $|\Pi| \leq k$ **do** , add (X_i, Π) to Ω
    end for
    Select a pair (X_i, Π) from Ω with the probability proportional to $exp(\frac{I(X_i,\Pi)}{2\Delta})$
    Add (X_i, Π) to $\mathcal{N}$; add X_i to V
end for
return $\mathcal{N}$

2. Use an $\epsilon/2$-differential private algorithm to generate a set of conditional distributions of D, such that for each AP pair (X_i,Π_i) in $\mathcal{N}$, we have a noisy version of the conditional distribution $P[X_i|\Pi_i]$. (We denote this noisy distribution as $P^*[X_i|\Pi_i]$.)

Algorithm 2 Noisy distribution

Input: D, k, $\mathcal{N}$,
Output: $\mathcal{P}^*$

Initialize $\mathcal{P}^* = \emptyset$
for $i = k + 1 \rightarrow d$ **do**
    Materialize the joint distribution $P[X_i, \Pi_i]$
    Generate differential private $P^*[X_i, \Pi_i]$ by adding Laplacian noise satisfying $Lap(\frac{2(d-k)}{n\epsilon})$
    Set negative values in $P^*[X_i, \Pi_i]$ to 0 and normalize
    Derive $P^*[X_i|\Pi_i]$ from $P^*[X_i, \Pi_i]$; add it to $\mathcal{P}^*$
end for
for $i = 1 \rightarrow k$ **do**
    Derive $P^*[X_i|\Pi_i]$ from $P^*[X_{k+1}, \Pi_{k+1}]$; add it to $\mathcal{P}^*$
end for
return $\mathcal{P}^*$

3. Use the Bayesian network $\mathcal{N}$ (constructed in the first phase) and the d noisy conditional distributions (constructed in the second phase) to derive an approximate distribution of the tuples in D, and then sample tuples from the approximate distribution to generate a synthetic dataset D^*.

3. Solution

As we mentioned, there are two problems with PrivBayes:

1. The selection of the first node in PrivBayes is random, and the mutual information between other nodes and the first node is used as the score function to construct the Bayesian network. These may result in candidate AP pairs with low mutual information joining the Bayesian network too early, which causes the average mutual information provided by all selected AP pairs to decrease, and ultimately there is a problem of poor quality of the constructed Bayesian network.

2. PrivBayes directly allocates the privacy budget without considering the different privacy needs among different individuals and attributes and thus may lead to unreasonable addition of noise, which has an impact on the usability and privacy of data.

In this section, we will present an overview of HPPrivBayes, our solution for releasing a high-dimensional dataset D satisfying ϵ-differentially privacy. Before we introduce HPPrivBayes, we will first introduce several definitions related to the HPPrivBayes algorithm.

3.1. HPPrivBayes-related Definitions

To solve the problem of poor quality of the Bayesian network constructed by PrivBayes, HPPrivBayes will use the average mutual information to determine the order in which the nodes enter the Bayesian network, increasing the amount of mutual information captured when using the exponential mechanism to select the AP pairs, which will in turn positively affect the quality of the network structure. It is worth noting that during the iteration of the algorithm, the order in which nodes enter only reflects the importance of the nodes in the data and does not directly leak the mutual information between the nodes. The following is the definition of average mutual information.

Definition 10 (Average Mutual Information (AMI)) Average mutual information measures the average degree of association of a variable with the others in a dataset. Given a dataset D, $\mathcal{S} = \{S_1, S_2, ..., S_d\}$ is the attribute set of D, the average mutual information of any attribute S_i with another attribute S_j is given by

$$AMI(S_i) = \frac{\sum_{j=1}^{d} I(S_i, S_j)}{d-1}.$$

Note that because the mutual information of one attribute with itself is not meaningful, we set $I(S_i, S_i) = 0$ for any $i \in [1, d]$.

Since different values in each attribute may also have different privacy requirements because of their different sensitivity, we would like to define a sensitivity weight that maps each

value in the same attribute to a score, which could help us evaluate the sensitivity of these values. Sensitivity weight should have the following characteristics:

- Non-negative: All private data always has a certain sensitivity, once its sensitivity is 0, the data no longer belongs to the category of private data, and it does not need privacy protection.
- Monotonous: The higher the sensitivity of a value, the greater the sensitivity weight of the value.
- Negative correlation: The less frequently a sensitive value shows in the dataset, the more information it contains, thus, the greater sensitivity weight it has.

The last property may require an example to explain. For example, in a medical dataset, there are fewer cases of *cancer* than *cold* in the attribute disease, the *cancer* case should be more sensitive (which is also true from a common sense perspective) as the number of people with this disease is relatively small, an attacker is more likely to combine information revealed by other attributes to identify the owner of the information, resulting in the disclosure of sensitive personal information. Therefore, the value *cancer* in the attribute disease requires more privacy budget than the value *cold*.

Definition 11 (Sensitivity Weight). Let SD_i be the sensitivity of the value s_i, and $P(s_i)$ the frequency with which s_i shows in the attribute S (i=1,...,n). Then the sensitivity weight of s_i is given by

$$w_i = \frac{SD_i}{\sum_{j=1}^{n} SD_j},$$

where $SD_i = -log_2(P(s_i))$.

Proof.

1. Non-negative: When $P(s_i) \in (0, 1]$, we have $log_2(P(s_i)) \in (-\infty, 0]$ according to the property of $y = log_2(x)$. Therefore, $SD_i = -log_2(P(s_i))$ is non-negative, and thus, w_i is non-negative for any s_i. Note that when $P(s_i) = 1$, it means that this attribute only contains one value, and in such case, data of this attribute will not leak privacy as every tuple shares the same information. Therefore, this value does not require privacy protection and thus its sensitivity weight will be 0. This is consistent with the case where $w_i = 0$ when $P(s_i) = 1$.
2. Monotonous: Let s_m, s_n be two values in attribute S such that $SD_m > SD_n$. According to the formula of sensitivity weight, we have $w_m > w_n$.
3. Negative correlation: Let s_m, s_n be two values in attribute S such that $P(s_m) < P(s_n)$.

Since $y = -log_2(x)$ is monotonously decreasing in $(0, \infty)$, then we have $SD_m > SD_n$. Thus, $w_m > w_n$.

3.2. HPPrivBayes

Let $L_1, L_2, L_3 \in \mathbf{R}$ be positive such that $L_1 + L_2 + L_3 = 1$. HPPrivBayes runs in five steps:

Step 1. Construct a k-degree Bayesian network on the dataset D.

HPPrivBayes will use the average mutual information to determine the order in which the nodes enter the Bayesian network. We first calculate the value of AMI corresponding to each attribute, reorder all the attributes by the value of AMI from the largest to the smallest to get $\widetilde{\mathcal{A}}$, and then add all the attribute nodes to the Bayesian network $\mathcal{N}$ sequentially according to this order. Therefore, the algorithm discards the set V, which has a recording parameter of attribute nodes in Algorithm 1, and directly carries out the Bayesian network based on $\widetilde{\mathcal{A}}$. Note that $\widetilde{\mathcal{A}} = \{\widetilde{\mathcal{A}}_1, \widetilde{\mathcal{A}}_2, ..., \widetilde{\mathcal{A}}_d\}$, and $\{\widetilde{\mathcal{A}}\}_1^{i-1} = \{\widetilde{\mathcal{A}}_1, \widetilde{\mathcal{A}}_2, ..., \widetilde{\mathcal{A}}_{i-1}\}$.

Algorithm 3 Construct Bayesian network (HPPrivBayes)

Input: D, k,
Output: $\mathcal{N}$

1: Initialize $\mathcal{N} = \emptyset$
2: Calculate AMI for all attributes according to Definition 10, reorder all the attributes by the value of AMI in a descending order to get $\widetilde{\mathcal{A}}$
3: Add $(\widetilde{\mathcal{A}}_1, \emptyset)$ to $\mathcal{N}$
4: **for** $i = 2 \to d$ **do**
5: Initialize $\Omega = \emptyset$
6: **for** each possible $\Pi \subset \{\widetilde{\mathcal{A}}\}_1^{i-1}$ and $|\Pi| \leq k$ **do** , add $(\widetilde{\mathcal{A}}_i, \Pi)$ to Ω
7: **end for**
8: Select a pair $(\widetilde{\mathcal{A}}_i, \Pi)$ from Ω with a probability proportional to $exp(\frac{\epsilon_1 I(D,(\widetilde{\mathcal{A}}_i,\Pi))}{2(d-1)\Delta I})$ and add it to $\mathcal{N}$
9: Add $(\widetilde{\mathcal{A}}_i, \Pi)$ to $\mathcal{N}$
10: **end for**
11: **return** $\mathcal{N}$

The privacy budget that we allocate to this step is $\epsilon_1 = L_1\epsilon$. Since the selection of the first node does not consume the privacy budget, for step 8 in Algorithm 3, the privacy budget consumed is $\epsilon_1/(d-1)$ for each selection of AP pairs using the exponential mechanism, and each time AP pairs are selected from the set Ω with probability proportional to $exp(\frac{\epsilon_1 I(D,(\widetilde{\mathcal{A}}_i,\Pi))}{2(d-1)\Delta I})$ and added to $\mathcal{N}$ until all nodes are added to $\mathcal{N}$. By the Sequential Composition property of differential privacy mentioned in Definition 6, Algorithm 3 satisfies ϵ_1-differential privacy.

Step 2. Divide all the attributes in dataset D into two groups, Group A for the more sensitive attributes and Group B for the less sensitive attributes.

Here, we calculate the mutual information between a sensitive attribute S and all others.

Then, we target the attributes whose mutual information is greater than or equal to the threshold value θ as more sensitive attributes and those whose mutual information is less than the threshold value θ as less sensitive attributes. For more sensitive attributes, we allocate $\epsilon_2 = L_2\epsilon$ privacy budget, and for less sensitive attributes, we allocate $\epsilon_3 = L_3\epsilon$ privacy budget. Let $r = L_2/L_3$.

Algorithm 4 Classification of Sensitive Attribute

Input: D, S, θ
Output: A, B

Compute the distribution for each attribute, including S
Calculates the joint distribution between S and each attribute
for $i = 1 \to d-1$ **do**
    $I(X_i, S) = \sum_{x_j \in X_i} \sum_{s \in S} p(x_j, s) \cdot log \frac{p(x_j,s)}{p(x_j)p(s)}$
    if $I(X_i, S) \geq \theta$ **then**
        $X_i \to A$
    else
        $X_i \to B$
    end if
end for
return A, B

Step 3. Let l be the number of less sensitive attributes, d the number of attributes in the dataset, k the degree of the Bayesian Network we construct, and $\epsilon_2 = L_2\epsilon$. For more sensitive attributes in Group A, each attribute will consume $\epsilon_2/(d-k-l)$ privacy budget. Thus, we inject noise satisfying $Lap(2(d-k-l)/(n\epsilon_2))$ into sensitive attributes using Laplace mechanism. This makes the more sensitive attributes meet ϵ_2-differential privacy.

Algorithm 5 Privacy Protection for attributes in Group A

Input: D, A, k, $\mathcal{N}$, ϵ_2
Output: $\mathcal{P}^*$

Initialize $\mathcal{P}^* = \emptyset$
for $X_i \in A$ **do**
    Materialize the joint distribution $P[X_i, \Pi_i]$
    Generate differential private $P^*[X_i, \Pi_i]$ by adding noise satisfying $Lap(\frac{2(d-k-l)}{n\epsilon_2})$
    Set negative values in $P^*[X_i, \Pi_i]$ to 0 and normalize
    Derive $P^*[X_i|\Pi_i]$ from $P^*[X_i, \Pi_i]$; add it to $\mathcal{P}^*$
end for
return $\mathcal{P}^*$

Step 4. For less sensitive attributes, the privacy budget for each of them is $\epsilon_3 = L_3\epsilon$, then we allocate different privacy budgets for different values in each attribute, according to the sensitivity weights of the values w_i which was mentioned in 3.2. That is, each value in each attribute in Group B consumes $w_i\epsilon_3$, and the added noise satisfies $Lap(2l/(nw_i\epsilon_3))$. Thus, this makes the less sensitive attributes meet ϵ_3- differential privacy.

Algorithm 6 Privacy Protection for attributes in Group B

Input: D, B, k, $\mathcal{N}$, ϵ_2
Output: $\mathcal{P}^*$

Initialize $\mathcal{P}^* = \emptyset$
for $X_i \in A$ **do**
    Materialize the joint distribution $P[X_i, \Pi_i]$
    for $x_j \in X_i$ **do**
        Compute the sensitivity weight w_j for each value x_j in attribute X_i
        Generate differential private $P^*[X_i, \Pi_i](x_i)$ by adding noise satisfying $Lap(\frac{2l}{nw\epsilon_3})$
        Set negative values in $P^*[X_i, \Pi_i]$ to 0 and normalize
        Derive $P^*[X_i|\Pi_i]$ from $P^*[X_i, \Pi_i]$; add it to $\mathcal{P}^*$
    end for
end for
return $\mathcal{P}^*$

Step 5. Use the Bayesian network $\mathcal{N}$ (constructed in Step 1) and the d conditional distributions (constructed in Step 3 and 4) to derive an approximate distribution of the tuples in D, and then sample tuples from the approximate distribution to generate a synthetic dataset D^*.

The flow chart of HPPrivBayes is shown in the figure below.

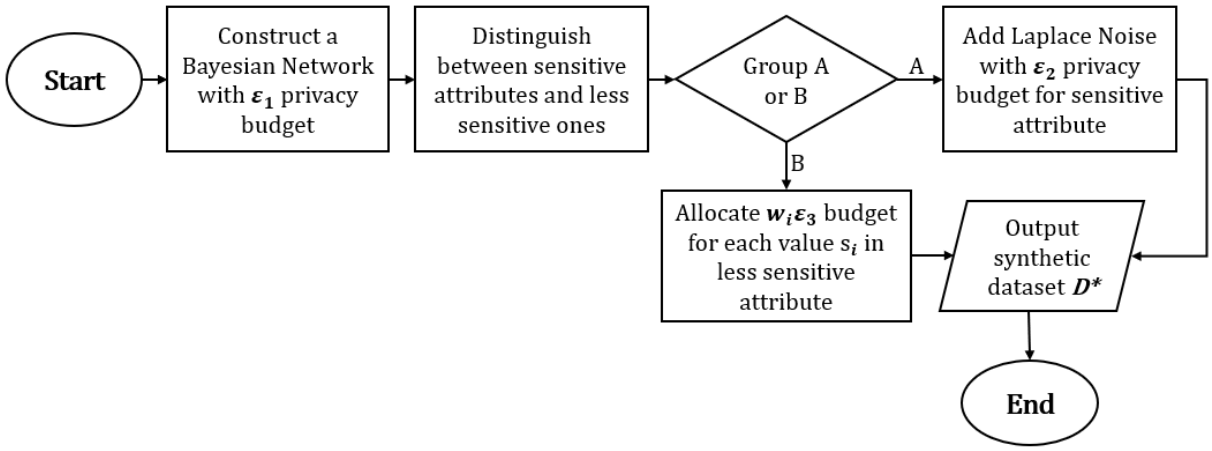


Figure 2: Flow Chart of the Privacy Allocation Method in HPPrivBayes

Lemma 1 The sensitivity of $P[X_i, \Pi_i]$ is 2/n.

Theorem 1 HPPrivBayes satisfies ϵ-Differential Privacy.

Proof.

1. In step 1, building a Bayesian network consumes $\epsilon_1 = L_1 * \epsilon$ privacy budget.
2. In step 2, we sample the dataset, calculate the mutual information of the *occupation* attribute and the other attributes, and divide the attributes in the dataset into two groups

with the threshold θ as the boundary. There is no privacy protection involved in this step as we consider the data publisher safe, so no privacy budget is used in this step.

3. In step 3, there will be $(d-l)$ attributes X_i in Group A. As we mentioned, we generate differentially private $P^*[X_i, \Pi_i]$ by adding Laplacian noise to $P[X_i, \Pi_i]$. For $(d-l)$ attributes, we will construct $(d-k-l)$ noisy conditional distributions. So if we allocate $\epsilon_2/(d-k-l)$ budget for each distribution, that is adding noise satisfying $Lap(2(d-k-l)/(n\epsilon_2))$ to $P[X_i, \Pi_i]$ since $P[X_i, \Pi_i]$ has sensitivity 2/n according to Lemma 1, we can ensure that the generation of $P^*[X_i, \Pi_i]$ satisfies $(\epsilon_2/(d-k-l))$-differential privacy. Thus, the whole step consumes $\epsilon_2/(d-k-l) * (d-k-l) = \epsilon_2$ privacy budget according to the composability property of differential privacy mentioned in 2.1.

4. In step 4, there will be l attributes X_i in Group B. For these attributes, we construct l noisy conditional distributions. So if we allocate ϵ_3/l budget for each attribute in Group B, the total consumption of privacy budget in this step will be ϵ_3 according to the composability property of differential privacy. And considering the process of assigning different privacy budgets to different values in each attribute, if a value x_j in attribute X_i with sensitivity weight w_j consumes $w_j * \epsilon_3/l$, then all the values in X_i will consume $\sum_j w_j\epsilon_3/l = \epsilon_3/l$, which satisfies the budget we allocate for every attribute in Group B.

5. In step 5, we generate a new dataset using the distributions we constructed, and there is no privacy protection involved in this step so no privacy budget is used in this step.

In general, HPPrivBayes satisfies $\epsilon = \epsilon_1 + \epsilon_2 + \epsilon_3$ differential privacy.

4. Experiments

4.1. Experimental Settings

4.1.1. Experiment Environment

Hardware environment: Intel(R) Core(TM) i7-1065G7 CPU @ 1.30GHz 1.50 GHz, memory 32G, operating system WIN11 64-bit; Software environment: Python programming language; Development tools: Pycharm.

4.1.2. Dataset

We will use the US Census dataset *Adult* for the experiment, which contains data on 45,222 people in the 1994 U.S. Census, with 15 attributes including age and education. For the experiment, we will use two-thirds of all the data as the training set and the remaining one-third as the test set and select 8 discrete and 6 continuous attributes.

Table 1: US Census dataset *Adult*

Attribute	Type	Size
age	continuous	
workclass	discrete	7
fnlwgt	continuous	
education	discrete	16
education-num	continuous	
marital-status	discrete	7
occupation	discrete	14
relationship	discrete	6
race	discrete	5
sex	discrete	2
capital-gain	continuous	
capital-loss	continuous	
hours-per-week	continuous	
native-country	discrete	41

To construct a Bayesian network, discretization of the dataset is required. Besides, to facilitate the subsequent stratification of sensitive values, we re-classify the native-country attribute into two categories, namely, US and non-US. The following is the number of attribute values after discretization.

Table 2: number of attribute values after discretization

Attribute	Size
age	5
workclass	7
fnlwgt	7
education	16
education-num	5
marital-status	7
occupation	14
relationship	6
race	5
sex	2
capital-gain	6
capital-loss	2
hours-per-week	7
native-country	2

4.1.3. Evaluation Indexes

The experiment part will involve three methods, the sum of mutual information, the Support Vector Machine (SVM) algorithm, and the α-marginal distribution. The sum of mutual information of a Bayesian network can show its quality. A better Bayesian network retains information better. The result of SVM can show the accuracy of classification and thus the accuracy of the established Bayesian model. SVM classification generally yields higher classification accuracies on better Bayesian network models. The α-marginal distribution can evaluate the similarity between the original dataset D, and the synthetic dataset.

According to the research of Zhang et al. (Zhang et al., 2017), it has been proved that we can convert the optimization problem of the Bayesian network constructed into the mutual information catch problem, that is, the more the sum of mutual information caught by the Bayesian network, the higher the quality of the Bayesian network. Therefore, in the first part of the experiment, the sum of mutual information in the Bayesian network is used as an evaluation index to prove that the Bayesian networks constructed by the HPPrivBayes method have higher qualities than those built by PrivBayes.

SVM is a machine learning algorithm first proposed by Cortes and Vapnik in 1995 (Cortes and Vapnik, 1995). It is a kind of binary classifier under supervised learning, available in both linear separable cases and non-linear indivisible cases. In the case of non-linearity, SVM can map the original feature space to a higher dimensional space by using nonlinear functions, so that we can find a hyperplane to segment the sample. Thus, nonlinear problems can be transformed into linear separable problems. The principle of segmentation is to maximize the interval. Commonly, we define the kernel function as the inner product of the mapping function

to avoid complex computations. There are many classic options for kernel functions, and we will use the Radial Basis Function (RBF) kernel in this article. We will train classifiers on both the original dataset and the generated datasets, each of which uses all other attributes to predict the classification of the target attribute chosen. For this sorting task on *Adult*, we choose the attribute *education* as the target attribute, to predict whether an individual has a secondary education or not. After learning and classifying datasets respectively, we evaluate their usabilities by comparing their classification accuracies. Let TP be the positive example that is judged correctly, TN be the negative example that is judged correctly, FP be the positive example that is judged wrongly, and FN be the negative example that is judged wrongly. The formula for classification accuracy is as follows:

$$\textit{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}.$$

For the α-marginal distribution, it is used to evaluate the accuracy of the α-marginal distribution of the overall generated data. In the experiments, the 2-marginal distribution and 3-marginal distribution were selected as examples. To measure the accuracy of the marginal distribution, the next section will calculate the average variation distance (Tsybakov, 2008) between the α-marginal distribution of the original dataset D and the α-marginal distribution of the generated datasets D_1 and D_2 respectively. The formula for total variation distance is as follows:

$$||\mu - \nu||_{TV} = \frac{1}{2}\sum_{x \in E} |\mu(x) - \nu(x)|.$$

4.2. Experiment Results and Analysis

In 4.2.1, we will compare the sums of mutual information of the Bayesian network constructed by HPPrivBayes with the ones constructed by PrivBayes. In 4.2.2 and 4.2.3, to compare the performance of the two algorithms PrivBayes and HPPrivBayes under different privacy budges ϵ, we will test the results of SVM and α-marginal distribution on the two algorithms PrivBayes and HPPrivBayes respectively. Note that in these two parts, the value of r in HP-PrivBayes will be set as 1/3. In this part, we will set ϵ to 0.2, 0.4, 0.6, 0.8, 1. Then, in 4.2.4, we will focus on the privacy allocation part of the HPPrivBayes and investigate its properties by adjusting the value of the budget allocation ratio parameter r of Group A to B in the method. For this part, we will set r to 1/6, 1/5, 1/4, 1/3, and 1/2, and compare the results of the SVM.

For all experiments in this paper, to get more accurate results, each experiment will be repeated 20 times and the average of the results will be used as the final result for plotting. For the HPPrivBayes method in all these experiments, we will use attribute *occupation* as the

sensitive attribute S to divide the other attributes into more sensitive and less sensitive two groups, and we will set θ as 0.1.

4.2.1. Bayesian network Quality Assessment

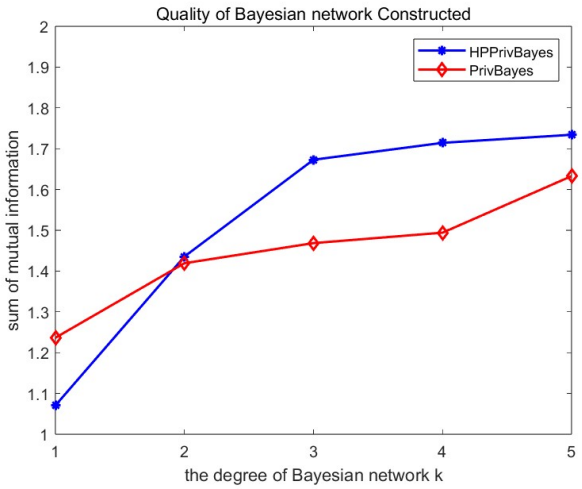


Figure 3: Bayesian network Quality comparison between HPPrivBayes and PrivBayes

Since the HPPrivBayes method improves on PrivBayes by selecting the first node of the network according to the average mutual information, we find that the ability of the Bayesian network to capture mutual information is not sensitive to the choice of the first node when k=1. Besides, it can also be seen that when k is larger than 1, HPPrivBayes builds better Bayesian networks than PrivBayes. It shows that the HPPrivBayes method is more advantageous in constructing Bayesian networks when k is higher than 1. Given that in practice, we usually make k greater than 1, we can consider the optimization of HPPrivBayes over PrivBayes for Bayesian network construction to be of practical application. Note that for all Bayesian networks we build in the following experiments, their degree will be k=4.

4.2.2. Algorithm Performance based on SVM

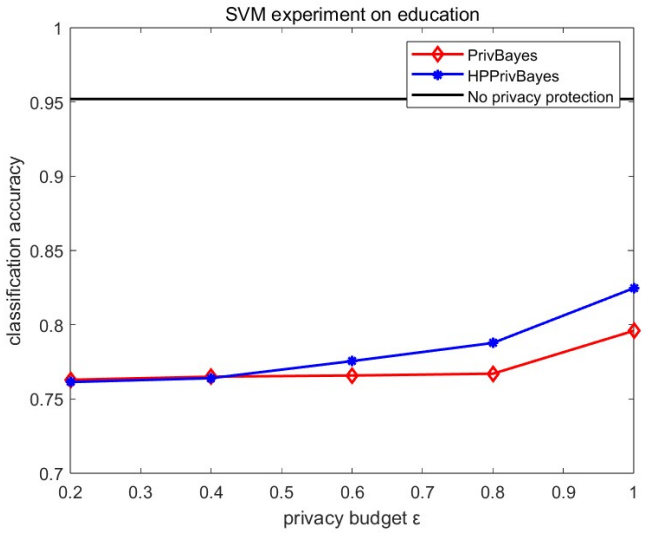
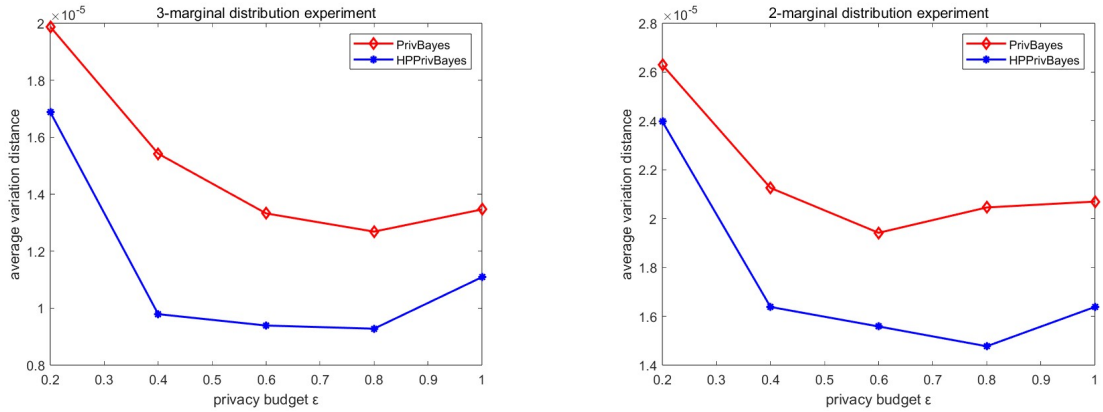


Figure 4: Classification result of SVM experiment using education as target attribute

It can be seen in Figure 4 that the classification accuracy of HPPrivBayes is higher than that of PrivBayes, which indicates that our changes over the PrivBayes method's privacy budget allocation improve the accuracy of the generated datasets. To explain this, this is because HP-PrivBayes can construct networks with higher quality as mentioned in 4.2.1, and it has a more rational allocation of the privacy budget and thus makes the overall usability of the generated data set increase. However, due to the differential privacy protection of the data, the accuracies of the dataset generated by PrivBayes and HPPrivBayes are significantly reduced compared to the original data, which can be observed from the graph.

Note that the classification accuracy increases as the privacy budget becomes larger. This is because when ϵ becomes larger, there will be less noise added to the synthetic dataset which reduces the difference between the generated data set and the original data set. Thus, the classification accuracy will increase. Besides, it can be seen that the curves of PrivBayes and HP-PrivBayes are close to each other when $\epsilon = 0.2$ and $\epsilon = 0.4$. The possible reason for this situation is that when the privacy budget ϵ becomes smaller, the generated data set is less sensitive to the personalized privacy allocation method due to the large amount of noise added.

4.2.3. Algorithm Performance based on α-marginal distribution



(a) 3-Marginal Distribution (b) 2-Marginal Distribution

Figure 5: Result of α-Marginal Distribution experiment on PrivBayes and HPPrivBayes

According to Figure 5, the average variation distance of the dataset generated by HP-PrivBayes from the original dataset is lower than that of the data generated by PrivBayes. This indicates that our differential privacy algorithm better guarantees the approximation of the statistical distribution of the generated dataset to the original dataset as compared to PrivBayes. Thus, our method's improvement on Bayesian network construction and personalized privacy budget allocation ensure the usability of the data better than PrivBayes while satisfying personalized privacy protection requirements.

Note that in both two graphs, all curves have a downward trend as the privacy budget becomes larger. This is because when ϵ becomes larger, there will be less noise added to the synthetic dataset. Thus, the difference between the generated data set and the original data set will reduce and the average variation distance from the generated dataset to the original dataset will decrease.

4.2.4. Test on r

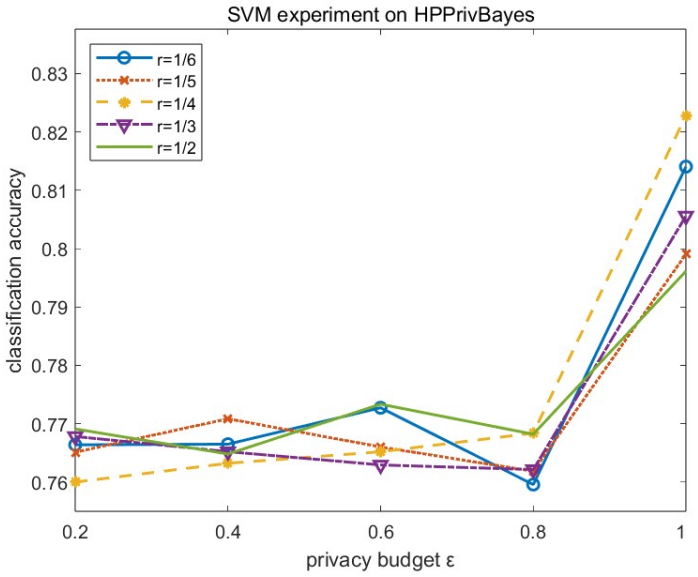


Figure 6: Result of SVM experiment on HPPrivBayes with different r

As can be seen from Figure 6, the closer r is to 1/6, i.e., the smaller r is, the greater the fluctuation of classification accuracy with the change in total privacy budget allocation. Note that the r represents the ratio of privacy budget allocation between groups A and B. A smaller r corresponds to the fact that we add a large amount of noise to group A while adding almost no noise to group B. The result of the experiment is consistent with our expectations because tilting too much protection budget to Group A without protecting Group B will lead to overprotecting the data of Group A. This will cause a decrease in the usability of the overall dataset, at which point the dataset may already be quite different from the original dataset even though the accuracy computed seems to be high. In such cases, the classification accuracies obtained in the experiment are very sensitive to the overall privacy budget. Therefore, the fluctuation amplitude of the curves under smaller r's will be larger.

In conclusion, the value of r could significantly impact the model's effect, the choice of r should be careful and may require some pre-experiments.

5. Conclusions

Most existing research directly sets the privacy requirements of each data and each attribute in the dataset as the same. However, in real life, different sensitivities of individuals and attributes in datasets lead to different privacy requirements. To not only satisfy the personalized privacy allocation but also ensure the publication of high-dimensional data that meets the differential privacy, this paper proposes a hierarchical personalized differential privacy algorithm (HPPrivBayes) based on PrivBayes, which is a classic data publishing algorithm that satisfies differential privacy. In this method, the average mutual information is used to determine the order of nodes to enter the Bayesian network, which solves the problem that the quality of the constructed Bayesian network is low and unstable due to the random selection of the first node of PrivBayes. For the privacy budget allocation part of HPPrivBayes, the privacy requirements of each attribute are divided into two levels according to the attributes' mutual information with sensitive attribute S, and the corresponding privacy protection degree will be matched for different levels of needs. Besides, the importance of different values in each less sensitive attribute will also be considered, and the corresponding privacy budget will be allocated according to their sensitivity weights.

We have seen that data released this way has both relatively high usability and accuracy. The optimization of HPPrivBayes over PrivBayes for Bayesian network construction improves the quality of constructed networks successfully under most practical scenarios. And the personalized allocation of privacy budgets in HPPrivBayes not only satisfies the different requirements of different individuals and attributes but also maintains the usability of synthetic datasets at a relatively high level, especially when under loose privacy protection. As the value of r could significantly impact the model's effect, the choice of r should be careful and may require some pre-experiments.

In the subsequent research, the algorithm's performance can be further optimized by using more effective algorithms to build Bayesian networks. As we found that even though HPPrivBayes gets better results than PrivBayes in all experiments with all privacy budgets, it does not work very well for strong privacy protection, that is when ϵ is small, perhaps further optimizations can be made to make HPPrivBayes perform better even with a stronger privacy budget allocated. Besides, further improvements may also be made to avoid the algorithm being too sensitive to the total privacy budget under small r.

Acknowledgements

As my four years of undergraduate study draws to a close, I am compelled to extend my heartfelt gratitude to several entities who have been instrumental in shaping my academic journey.

First and foremost, I am deeply indebted to my beloved alma mater, Jinan University, and the University of Birmingham, for furnishing me with invaluable platforms for continuous learning and personal growth. Their commitment to excellence has been a guiding light throughout my academic endeavors.

To my family, whose unwavering support both materially and spiritually has been the bedrock of my educational pursuits, I extend my sincerest appreciation. We have been through some trials and tribulations together in the past and we will undoubtedly continue to work hand in hand together in the future.

I am also immensely grateful to my teachers and classmates for their unwavering support, both inside and outside the classroom. Their guidance, encouragement, and camaraderie have enriched my university experience immeasurably. Their influence will undoubtedly remain cherished memories, serving as a steadfast foundation upon which I will continue to improve in the future.

Besides, I want to extend my sincere gratitude to my favorite football team for their matches providing much-needed relaxation during my graduation project. Their relentless spirit and pursuit of excellence inspired me to strive for perfection in my academic endeavors.

Lastly, I wish to express my heartfelt thanks to my thesis advisor Associate Professor Dr. Gan, and my groupmates again for their invaluable assistance throughout my graduation project. Your support and guidance have been indispensable, and for that, I am profoundly grateful.

References

Dima Alhadidi, Noman Mohammed, Benjamin CM Fung, and Mourad Debbabi. Secure distributed framework for achieving ε-differential privacy. In *Privacy Enhancing Technologies: 12th International Symposium, PETS 2012, Vigo, Spain, July 11-13, 2012. Proceedings 12*, pages 120–139. Springer, 2012.

Siyang Chen. Personalized privacy data publishing method based on improved bayesian network. *Software Guide*, 20:213–216, 2021. ISSN 1672-7800.

Corinna Cortes and Vladimir Vapnik. Support vector machine. *Machine learning*, 20(3):273–297, 1995.

Cynthia Dwork. Differential privacy. In *International colloquium on automata, languages, and programming*, pages 1–12. Springer, 2006.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer, 2006.

Zhen Gu, Guoyin Zhang, and Chen Yang. Horizontally partitioned data publication with differential privacy. *Security and Communication Networks*, 2022, 2022.

Mingzhu Li and Xuebin Ma. Bayesian networks-based data publishing method using smooth sensitivity. In *2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*, pages 795–800. IEEE, 2018.

Xiaotian Lu, Chunhui Piao, Xingyu Yang, and Yingjie Bai. Research on differential privacy high dimensional data publishing technology via bayesian networks. *Computer Engineering*, pages 1–14, 2023. ISSN 1000-3428. doi: 10.19678/j.issn.1000-3428.0067967.

Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3–es, 2007.

Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 94–103. IEEE, 2007.

Frank D McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pages 19–30, 2009.

Guanglei Meng, Zelin Cong, Bin Song, Tingting Li, Chenguang Wang, and Minzhe Zhou. Review of bayesian network structure learning. *Journal of Beijing University of Aeronautics and Astronautics*, pages 1–24, 2023. ISSN 1001-5965. doi: 10.13700/j.bh.1001-5965.2023.0445.

Noman Mohammed, Shuang Wang, Rui Chen, and Xiaoqian Jiang. Private genome data dissemination. *Medical Data Privacy Handbook*, pages 443–461, 2015.

Chundi Mu, Jianbin Dai, and Jun Ye. Bayesian network for data mining. *Journal of software*, 11(5):660–666, 2000. ISSN 1000-9825. doi: 10.13328/j.cnki.jos.2000.05.012.

Ke Pan and Kaiyuan Feng. Differential privacy-enabled multi-party learning with dynamic privacy budget allocating strategy. *Electronics*, 12(3):658, 2023.

Haina Song. *Research on Key Technologies of Hierarchical Privacy Preservation in Data Collection and Publication*. PhD thesis, Beijing University of Posts and Telecommunications, 2021.

Latanya Sweeney. k-anonymity: A model for protecting privacy. *International journal of uncertainty, fuzziness and knowledge-based systems*, 10(05):557–570, 2002.

Peng Tang. *Research on Differentially Private Multi-Party Data Publishing*. PhD thesis, Beijing University of Posts and Telecommunications, 2019a.

Shiyi Tang. The research on data publication algorithms satisfy in differential privacy, 2018.

Yuwei Tang. Research on the optimization of bayesian differential privacy method for high-dimensional data. Master's thesis, Guangxi Normal University, 2019b.

A.B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer Series in Statistics. Springer New York, 2008. ISBN 9780387790527. URL `https://books.google.co.jp/books?id=mwB8rUBsbqoC`.

Liang Wang, Weiping Wang, and Dan Meng. Privacy preserving data publishing via weighted bayesian networks. *Journal of computer research and development*, 53(10):2343–2353, 2016. ISSN 1000-1239.

Biao Xiao, Hongqiang Yan, Haining Luo, and Jucheng Li. Research on improvement of bayesian network privacy protection algorithm based on differential privacy. *Netinfo Security*, 20:75–86, 2020. ISSN 1671-1122.

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

Wei Zhang, Jingwen Zhao, Fengqiong Wei, and Yunfang Chen. Differentially private high-dimensional data publication via markov network. *EAI Endorsed Transactions on Security and Safety*, 6(19), 2019.

Xing Zhang and Hao Chen. A research review of high-dimensional data publishing based on a differential privacy model. *CAAI transactions on intelligent systems*, 16(6):989–998, 2021. ISSN 1673-4785.

Dan Zhu. Bayesian network-based analysis of the causes of crowding in college cafeterias. In *Proceedings of 4th International Symposium on Economic Development and Management Innovation (EDMI 2022)*, pages 137–151. BCP Business and Management, 2022.