Hindawi
Security and Communication Networks
Volume 2021, Article ID 6684179, 13 pages
https://doi.org/10.1155/2021/6684179

Research Article

PLDP: Personalized Local Differential Privacy for Multidimensional Data Aggregation

Zixuan Shen, Zhihua Xia, and Peipeng Yu

School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China

Correspondence should be addressed to Zhihua Xia; xia_zhihua@163.com

Received 23 November 2020; Revised 24 December 2020; Accepted 2 January 2021; Published 28 January 2021

Academic Editor: Liguo Zhang

The collection of multidimensional crowdsourced data has caused a public concern because of the privacy issues. To address it, local differential privacy (LDP) is proposed to protect the crowdsourced data without much loss of usage, which is popularly used in practice. However, the existing LDP protocols ignore users' personal privacy requirements in spite of offering good utility for multidimensional crowdsourced data. In this paper, we consider the personality of data owners in protection and utilization of their multidimensional data by introducing the notion of personalized LDP (PLDP). Specifically, we design personalized multiple optimized unary encoding (PMOUE) to perturb data owners' data, which satisfies ϵ_{total}-PLDP. Then, the aggregation algorithm for frequency estimation on multidimensional data under PLDP is developed, which is described in two situations. Experiments are conducted on four real datasets, and the results show that the proposed aggregation algorithm yields high utility. Moreover, case studies with four real datasets demonstrate the efficiency and superiority of the proposed scheme.

1. Introduction

In big data era, companies and institutions have noticed the big value of the data and are highly motivated to collect high-dimensional crowdsourced data to make data-driven decisions. The collection and analysis of data are beneficial to companies as well as the society; however, the data owners' privacy makes the biggest concern. In recent years, local differential privacy (LDP) [1, 2] has been found practical value in collection and utilization of data owners' data with the privacy preserved. In an LDP scheme, the data owners perturb their sensitive data before data outsourcing and then report the perturbed data to the server. In this way, the server cannot infer the owners' actual data with strong confidence, however, can still make the accurate estimation of data distribution as it was inferred from the unperturbed data. Considering the desirable properties of LDP, it has been adopted in practice and performs excellently. For example, Apple Inc. collects users' emoji records to discover the popular emojis [3] under LDP. Microsoft also designs the LDP scheme to collect application telemetry to improve user experience [4].

Although the existing LDP schemes are good solutions for data distribution estimation with privacy protected, they ignore data owners' personal privacy requirements. Specifically, in most existing LDP schemes [5, 6], all the owners perturb their different dimensions of data with the same privacy budget, which is set by the server, while it is the truth that different data have a different importance to each owner. In the real world, data owners must have different privacy requirements for their data. Furthermore, if the privacy protection level provided by the server is lower than an owner's need, the owner may be reluctant to share the data. Therefore, it is worth designing the LDP scheme with the personalized privacy allocation mechanism.

In this paper, we propose a multidimensional joint distribution estimation scheme with the personalized local differential privacy (PLDP), in which each data owner has his personal privacy requirement for each dimension of data. Specifically, the server sets an average privacy budget $\varepsilon_{\text{average}}$ to all of its data owners $O = \{o_1, o_2, \ldots, o_N\}$, and the owner o_i can split and allocate $m_i \times \varepsilon_{\text{average}}$ to each dimension of his m_i-dimensional data personally. Then, the data of different dimensions of the owner o_i will be perturbed with the

different privacy budgets. The data owners need not to report their privacy allocation to the server, i.e., the server only holds data owners' perturbed data and their total privacy budgets. Thus, a PLDP scheme provides stronger privacy assurance than the normal LDP scheme.

Moreover, a single data owner may not have the data of all the dimensions required by the server. In our scheme, an owner is allowed to report the data in a part of the dimensions. But, in this way, it is likely that the record of some high-dimensional data could not exist. Then, it is difficult for the server to estimate such dimensions of data. To address it, an aggregation algorithm is developed for joint distribution estimation on multidimensional data under PLDP. The contributions can be summarized as follows:

(1) We propose a new privacy notion called personalized local differential privacy (PLDP), which allows personalized privacy protection for different inputs than LDP. It has higher security and is more personalized than traditional LDP.

(2) We design a perturbed mechanism personalized multiple optimized unary encoding (PMOUE) and develop the aggregation algorithm for frequency estimation on multidimensional data under PLDP to estimate the joint distribution of multidimensional data.

(3) We propose ε_w to measure the personalized privacy protection level of multidimensional data for a single data owner o_i.

(4) Experiments on four real datasets validate the efficiency and superiority of PLDP. Compared with LoPub, PLDP outperforms it with high security, high privacy protection, considerable data utility, and low time consumption.

1.1. Roadmap. In Section 2, we review previous work related to LDP. Then, we give the preliminaries of LDP in Section 3. In Section 4, we describe the problem statement, define the new notion PLDP, and compare it to the traditional LDP. Then, we design the perturbation mechanism PMOUE and develop the aggregation algorithm to estimate the joint distribution of multidimensional data under PLDP. Experimental results are shown in Section 5. Finally, we present the conclusions of this paper in Section 6.

2. Related Work

Differential privacy is a rigorous mathematical definition of privacy for securely sharing the statistic of a dataset on a server [7]. When a requester requests a statistic value of a dataset, the server will calculate the value and then send a disturbed but usable value to the requester. An algorithm is said to be differentially private if the requester cannot infer any single data in the dataset by analyzing the statistic values. In a DP scheme, the server is supposed to be trusted and has all data owners' raw data. However, in the big data era, the data on the server are collected from users. The collection of data provides much useful information to the public; however, the data owners' privacy becomes the big concern as the server generally cannot be fully trusted in the real world. Accordingly, the local differential privacy (LDP) schemes are proposed, in which the owners perturb their data locally and then send the perturbed version to the server. With the perturbed data from data owners, the server can still estimate the distribution of the data.

As the first application of LDP to a real-world problem, Erlingsson et al. [8] proposed RAPPOR to securely estimate the character frequency in a set of strings. Specifically, each data owner uses k hash functions to hash his string onto k Bloom filters [9]. Then, two randomized response (RR) mechanisms, i.e., permanent randomized response and instantaneous randomized response, are proposed to perturb the bits in k Bloom filters. After receiving the perturbed data from the data owners, the server combines the mapping matrix with the LASSO regression [10] to estimate the character frequency. Kim et al. [11] also used randomized response mechanism to collect indoor position records for estimating the density of the specified indoor area. Several fixed points are selected in the indoor area, and each position is represented by its nearest point and denoted as a binary string. Then, the binary string is perturbed by the RR mechanism and then reported. Some researchers tried to reduce the communication cost by reporting a randomly selected bit from the binary string. The bit is also perturbed by RR mechanism. Then, the maximum likelihood estimation [12] and LASSO regression [13] are utilized to the data frequency.

The schemes mentioned above are designed for one-dimensional data. Considering the multidimensional data generally contains more valuable information, several LDP schemes for the multidimensional data are proposed recently. Ren et al. proposed an LDP scheme, named LoPub [14], to synthesize the high-dimensional dataset with the similar distribution to the real dataset. The data are encoded by Bloom filters and perturbed by RR mechanism as in [8]. In order to decrease the computation complexity, the authors tried to calculate the joint distribution of the small set of attributes at first and then calculate the joint distribution of the all attributes by multiplication. Specifically, the attributes with high mutual correlation are clustered together, generating the attribute clusters. The attribute clusters are considered to be independent of each other. Then, the joint attribute distribution within the clusters is calculated by the expectation-maximization algorithm [15] and LASSO regression. The joint distribution of all attributes is obtained by multiplying the distribution of all clusters. Zhang et al. proposed an LDP scheme, named CALM [16], to estimate marginal tables of multidimensional data. Instead of directly generating all marginal tables, the authors constructed many subsets of attributes and chose the randomization algorithm according to the marginal sizes. Expectation-maximization algorithm is used to estimate the marginal distribution. Since some attributes should exist in different subsets, the authors considered the marginal distributions to provide more accurate estimation.

Some researchers considered that different data would have different privacy requirements. Gu et al. proposed an

input-discriminative LDP scheme, in which the server sets different privacy budgets to attribute values. An input-discriminative unary encoding is designed to perturb the attribute value with the specified budgets. In [17, 18], the server provides several privacy budgets and owner can choose one budget for himself to perturb his data. Then, both the perturbed data and the selected budget are reported to the server. The server will estimate the distributions by grouping the data according to the budget. In [19], it is the data owner who decides the privacy budget for his own data; however, it still needs to report the privacy budget to the server for frequency estimation. The schemes [17–19] have tried to consider the personal privacy requirement, but all of them are designed for one-dimensional data. In addition, the server needs to know the privacy budgets that are applied to the disturbed data to estimate the data distribution, which would also expose privacy to the server.

To sum up, there is still a gap in the research on the LDP schemes that support the personalized privacy requirement for multidimensional data. Our goal is to design a scheme where the data owners can choose and perturb their data personally and the server can make a good distribution estimation with such perturbed data.

3. Preliminaries

3.1. Differential Privacy. Differential privacy (DP) is a rigorous mathematical definition of privacy for securely sharing the statistic of a dataset on a server [7]. In a DP scheme, the data owners report their raw data to the trusted server. Then, the server sends a perturbed query result to the requester by adding noise, such as Laplace noise [20]. In this way, the requester is unable to infer much about any single data in the dataset. A formal definition of DP is presented as follows.

Definition 1. differential privacy (DP) [7]). For a given privacy budget $\varepsilon \in R^+$, a randomized mechanism M satisfies ϵ-DP if and only if any neighboring dataset D, D' differs in at most one record and all subsets Y ⊆ range (M) and it has

$$\frac{\Pr[M(D) \in Y]}{\Pr[M(D') \in Y]} \leq e^{\varepsilon}. \tag{1}$$

3.2. Local Differential Privacy. In DP schemes, the server is fully trusted and can hardly apply to the case of privacy-aware crowdsourced systems. Accordingly, the local differential privacy (LDP) scheme is proposed, in which the data owners perturb their data locally and then send the perturbed version to the server. After receiving the perturbed data from owners, the server can still estimate the distribution of the data without knowing much about the data of the single owner. The formal definition of DP can be described as follows.

Definition 2. local differential privacy (LDP) [2]). For a given privacy budget $\varepsilon \in R^+$, a randomized mechanism M satisfies ε-LDP if and only if for any pair of inputs x, x' and any possible output y∈Range (M), it has

$$\frac{\Pr[M(x) = y]}{\Pr[M(x') = y]} \leq e^{\varepsilon}. \tag{2}$$

Theorem 1. *Sequential composition of LDP [21]). If the randomized mechanism M_i satisfies ε_i-LDP, for the given privacy budgets $\varepsilon_i \in R_i^+$, $i = 1, 2, \cdots, k$. Then, their sequential combination $M = (M_1, M_2, \ldots, M_k)$ satisfies $(\sum_{i=1}^{k} \varepsilon_i)$-LDP.*

Proof. For any inputs x, x', and output y, we have

$$\begin{aligned}\Pr[M(x) = y] &= \Pr[M_1(x_1) = y_1]\Pr[M_2(x_2) = y_2], \ldots, \Pr[M_k(x_k) = y_k] \\ &\leq e^{\varepsilon_1}\Pr[M_1(x_1') = y_1]e^{\varepsilon_2}\Pr[M_2(x_2') = y_2], \ldots, e^{\varepsilon_k}\Pr[M_k(x_k') = y_k] \\ &= e^{\sum_{i=1}^{k}\varepsilon_i}\Pr[M(x') = y].\end{aligned} \tag{3}$$

According to Theorem 1, a given total privacy budget $\varepsilon_{\text{total}}$ can be split into ε_i with $\varepsilon_{\text{total}} = \sum_{i=1}^{k} \varepsilon_i$, where each ϵ_i denotes the privacy budget of a randomized mechanism M_i. □

3.3. Optimized Unary Encoding

3.3.1. Randomized Response. Randomized response (RR) [22, 23] is a mainstream perturbation mechanism for LDP. The main idea is to give a random answer to a sensitive question. Accordingly, RR will disturb an input x to an output y as

$$P(y \mid x) = \begin{cases} x, & \text{w.p. } p, \\ x', & \text{w.p. } 1 - p. \end{cases} \tag{4}$$

Specifically, the interviewee gives the genuine answer x with probability p and gives the opposite answer x' with probability 1-p. In this way, RR satisfies $\ln(p/(1-p))$-LDP.

Considering RR only works for binary data, some perturbation mechanisms generalize and optimize it, such as direct encoding (DE) [24], histogram encoding (HE) [25], unary encoding (UE) [8, 26], and local hashing (LH) [6, 27].

In UE, an input x is encoded as a length-l binary vector, with only the bit corresponding to x set to 1. Then, the binary vector will be perturbed bit by bit with probability p and q as follows:

$$\Pr(\mathrm{UE}(x[i]) = 1) = \begin{cases} x[i] = 1, & \text{w.p. } p, \\ x[i] = 0, & \text{w.p. } q. \end{cases} \tag{5}$$

If UE uses $p + q = 1$, it becomes symmetric unary encoding (SUE) [8]. If UE uses optimized choices of p and q, it becomes optimized unary encoding (OUE) [26].

3.3.2. Optimized Unary Encoding. Optimized unary encoding (OUE) [26] converts the input $x = i$ into a binary vector $v = (0, \ldots, 0, 1, 0 \ldots, 0)$ with length-l, where the i-th bit is 1. Then, v is perturbed as follows:

$$\begin{cases} \Pr(y[i] = 1 \mid v[i] = 1) = p = \dfrac{1}{2}, \\ \Pr(y[i] = 0 \mid v[i] = 1) = 1 - p = \dfrac{1}{2}, \\ \Pr(y[i] = 1 \mid v[i] = 0) = q = \dfrac{1}{e^{\varepsilon} + 1}, \\ \Pr(y[i] = 0 \mid v[i] = 0) = 1 - q = \dfrac{e^{\varepsilon}}{e^{\varepsilon} + 1}. \end{cases} \tag{6}$$

In this way, OUE satisfies ϵ-LDP.

Proof. For any pair of inputs x and x', we denote their corresponding vector as v and v', respectively, and denote the output vector as y. Then, we have $\theta = ((\Pr(y \mid v))/(\Pr(y \mid v'))) = \prod_{i=1}^{l} ((\Pr(y[i] \mid v[i]))/(\Pr(y[i] \mid v'[i])))$. According to the OUE, there is only one '1' bit in both the v and v'. Let us assume that $v[s] = v'[t] = 1$, and there are four different cases for θ since the vector y could have four different possible values, which is listed as follows:

$$\theta = \begin{cases} \theta_1 = \dfrac{\Pr(y[s] = 0 \mid v[s] = 1)\Pr(y[t] = 0 \mid v[t] = 0)}{\Pr(y[s] = 0 \mid v'[s] = 0)\Pr(y[t] = 0 \mid v'[t] = 1)}, & \text{if } y[s] = 0, y[t] = 0, \\ \theta_2 = \dfrac{\Pr(y[s] = 0 \mid v[s] = 1)\Pr(y[t] = 1 \mid v[t] = 0)}{\Pr(y[s] = 0 \mid v'[s] = 0)\Pr(y[t] = 1 \mid v'[t] = 1)}, & \text{if } y[s] = 0, y[t] = 1, \\ \theta_3 = \dfrac{\Pr(y[s] = 1 \mid v[s] = 1)\Pr(y[t] = 0 \mid v[t] = 0)}{\Pr(y[s] = 1 \mid v'[s] = 0)\Pr(y[t] = 0 \mid v'[t] = 1)}, & \text{if } y[s] = 1, y[t] = 0, \\ \theta_4 = \dfrac{\Pr(y[s] = 1 \mid v[s] = 1 \mid)\Pr(y[t] = 1 \mid v[t] = 0)}{\Pr(y[s] = 1 \mid v'[s] = 0)\Pr(y[t] = 1 \mid v'[t] = 1)}, & \text{if } y[s] = 1, y[t] = 1. \end{cases} \tag{7}$$

Then, according to Formula (6), we have

$$\theta = \begin{cases} \theta_1 = \dfrac{(1-p)(1-q)}{(1-q)(1-p)} = 1, \\ \theta_2 = \dfrac{(1-p)q}{(1-q)p} = \dfrac{q}{1-q}, \\ \theta_3 = \dfrac{p(1-q)}{q(1-p)} = \dfrac{1-q}{q}, \\ \theta_4 = \dfrac{pq}{qp} = 1, \end{cases} \tag{8}$$

and $q = (1/(e^{\varepsilon} + 1))$. Since ϵ is definitely a positive number, we have $q < (1/2)$. Then, it is derived that

$$\theta \le \theta_3 = \frac{1-q}{q} = e^{\varepsilon}, \tag{9}$$

which demonstrates that OUE satisfies ϵ-LDP. □

3.4. Least Absolute Shrinkage and Selection Operator Regression. For real multidimensional data, its joint distribution may be sparse. To estimate the joint distribution of real multidimensional data, the least absolute shrinkage and selection operator (LASSO) regression is usually used. Since when solving a multivariate objective function as follows,

$$\widehat{\beta} = \min_{\beta} \|y - X\beta\|_2^2, \tag{10}$$

where y is the label, X is the vector of an input sample, and $\beta = (\beta_1, \beta_2, \ldots, \beta_m)$ is the vector of regression coefficients; the overfitting problem may occur. To address it, LASSO regression adds L1-norm as a penalty term after the objective function. Then, constructing a penalty function as follows,

$$\beta^* = \min_{\beta} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1, \tag{11}$$

where λ is a data dependent parameter.

In this way, when updating the regression coefficient β_i to β_i^* with learning rate η, $\beta_i^* = \beta_i - \eta((\partial(\beta^*))/(\partial(\beta_i)))$. Since $(\partial(\lambda\|\beta\|_1)/\partial(\beta_i)) = \lambda$ or $-\lambda$, the updated β_i^* is likely to be zero. In this way, LASSO regression forces the sum of the absolute value of the regression coefficients to be less than a fixed value, which keeps certain coefficients to be set to zero and accordingly results in a simpler model that does not include those zero coefficients. As a result, LASSO regression is quite suitable to solve the sparse linear regression because it can compress the coefficients of variables and make some regression coefficients become zero.

4. PLDP: Personalized Local Differential Privacy

4.1. System and Threat Model. In the existing multidimensional LDP schemes, the server sets a fixed privacy budget to each attribute to facilitate the estimation of the joint distribution. Since the same attributes of all owners are assigned the same amount of privacy budget, it ignores the personal privacy requirement of owners. However, in real life, the privacy levels of attributes could be distinct to the data owners, i.e., each owner has his personal privacy requirements for his data record. Therefore, it could be a better way to assign different budgets, which are personally decided by the data owner, to each attribute. In this paper, we proposed a multidimensional LDP scheme where the data owner can assign the personal privacy budget to the data he plans to upload according to his personal privacy requirement.

4.1.1. System Model. This paper considers a server that would like to collect a set of data records with d attributes from various data owners. The attributes are denoted as $\{A_1, A_2, \ldots, A_d\}$. To preserve the privacy, the owners are allowed to perturb the data according to his personal privacy requirement before uploading the data. Our goal is to enable the server to estimate the joint distribution of attributes from the perturbed data to benefit the public. Specifically, our system model involves a server and a set of data owners $O = \{o_1, o_2, \ldots, o_N\}$, where the owner o_i submits a perturbed data record $X_i = (x_{i1}, x_{i2}, \ldots, x_{id})$ to the server. In practice, a data owner does not have to possess or be willing to upload the all of d-dimensions of data. Thus, our scheme allows the owner to upload his record with some dimensions of data empty. After receiving the data records, the server puts the owners that report the same dimensions of data into the same group. Then, the joint distribution of data will be conducted within the group or by aggregating several groups. The system model of the proposed scheme can be illustrated in Figure 1.

4.1.2. Threat Model. We assume that the server is untrusted, since the server may leak the data owners' privacy, such as being hacked, or selling owners' data to a third party for profit. In this way, the adversary is assumed to possess a perturbed database with d attributes collected by the server, the principle of the perturbation mechanism, and the total privacy budgets of all owners.

4.2. Definition of PLDP and Its Relationships with LDP. In this paper, we propose a new LDP notion, named personalized local differential privacy (PLDP). In a PLDP scheme, the server sets an average privacy budget $\varepsilon_{\text{average}}$ for all of the data owners. If an owner plans to report a data record with m elements unemptied, the owner will hold a total amount of privacy budget $\varepsilon_{\text{total}} = m \times \varepsilon_{\text{average}}$. Then, the owner can personally allocate $\varepsilon_{\text{total}}$ to m elements according to his privacy requirement. In addition, the privacy budget allocation will not be reported to the server, which enhances the security of the users' data.

Definition 3. personalized local differential privacy (PLDP)). For a given average privacy budget $\varepsilon_{\text{average}} \in R^+$, each data owner allocates the total privacy budget $\varepsilon_{\text{total}} = m \times \varepsilon_{\text{average}}$ personally to his data record with m unemptied elements. Denote the m separate privacy budgets as $\varepsilon_i > 0$, $i = 1, \ldots, m$, and it definitely has $\varepsilon_{\text{total}} = \sum \varepsilon_i$. Then, a randomized mechanism M satisfies $\varepsilon_{\text{total}}$-PLDP if and only if for any pair of records x', x'' with the same unemptied elements, and any output $y \in \text{Range}(M)$, we have

$$\frac{\Pr[M(x') = y]}{\Pr[M(x'') = y]} \leq e^{\varepsilon_1' + \varepsilon_2' + \cdots + \varepsilon_m'} = e^{\varepsilon_1'' + \varepsilon_2'' + \cdots + \varepsilon_m''} = e^{\varepsilon_{\text{total}}}, \tag{12}$$

where ε_i' and ε_i'' are the privacy budgets allocated to x' and x'', respectively.

4.2.1. Relationships with LDP. In a multidimensional data scenario, if the privacy budgets for each attribute are the same, i.e., $\varepsilon_1' + \varepsilon_2' + \cdots + \varepsilon_m' = \varepsilon_1'' + \varepsilon_2'' + \cdots + \varepsilon_m''$, the PLDP becomes LDP. It means PLDP is a generalized version of LDP. The relationship between LDP and PLDP can be specified by the following two theorems.

Theorem 2. *For any pair of records x', x'' with m attributes, if a perturbation mechanism M satisfies ε-LDP, it also satisfies ε_{total}-PLDP with $\varepsilon_{total} = \varepsilon$.*

Proof. In a common LDP scheme without considering the personal privacy requirement of data owner, the privacy budget of each attribute equals to (ε/m). It can be considered as having all the ε_i equal to (ε/m) in $\varepsilon_{\text{total}}$-PLDP with $\varepsilon_{\text{total}} = \sum_{i=1}^{m} \varepsilon_i = \varepsilon$.

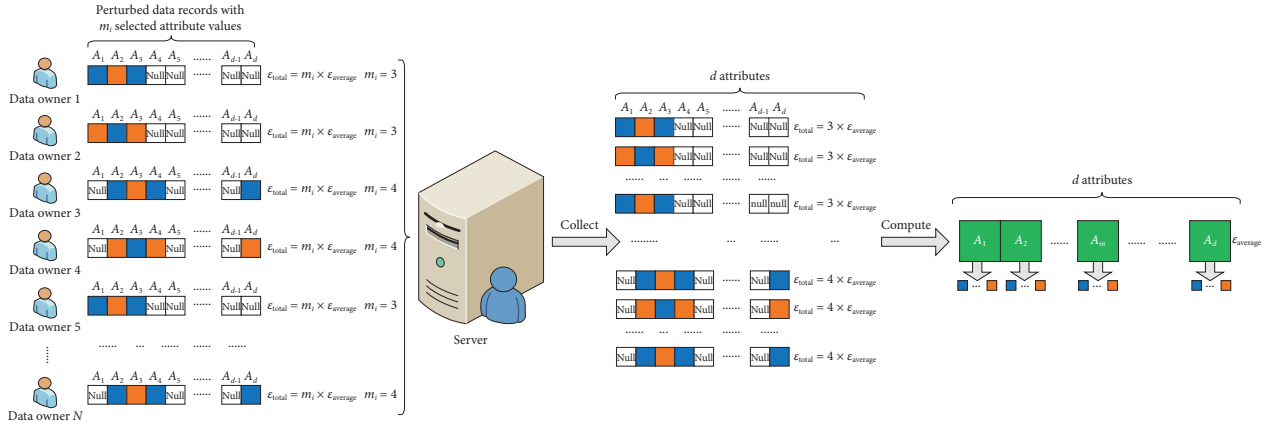

Figure 1: The architecture of the system model.

Theorem 2 indicates that LDP is a special case of PLDP. The PLDP can be considered as a generalization of LDP. □

Theorem 3. *For any pair of records x', x'' with m attributes, if a perturbation mechanism M satisfies ε_{total}-PLDP, it also satisfies ε-LDP with $\varepsilon = m \times min\ (\varepsilon_i) \le \varepsilon_{total}$, $i = 1, \ldots, m$, where ε_i denotes the privacy budget of i-th attribute.*

Proof. In a common LDP scheme for the multidimensional data record, each attribute is put on the same privacy budget. To guarantee the privacy, we should put the smallest privacy budget min (ε_i) to all of the attributes to get an LDP scheme with $\varepsilon = m \times \min\ (\varepsilon_i) \le \varepsilon_{\text{total}}$ under the same perturbation mechanism. □

4.2.2. Security Comparison. In a PLDP scheme, the privacy budget allocation is also a kind of privacy and thus not reported to the server. In addition, in a PLDP scheme, the owner can freely allocate his privacy budget to the attributes. When a small budget is put on an attribute, it means the attribute is very sensitive to the data owner and the privacy budget on this attribute also weighs a lot. Nevertheless, if a large budget is put on an attribute, it means the attributes are not so important and the privacy budget on this attribute may not weigh much. Here, we definite a notion to quantize the amount of weighted privacy budget, which can be regarded as the real privacy budget to a data owner in the PLDP scheme.

Definition 4 (weighted privacy budget ε_w). With a given total privacy budget $\epsilon_{\text{total}} = m \times \varepsilon_{\text{average}}$ from the server, the data owner splits it into m parts denoted as ε_i, $i = 1, \ldots, m$. Then, the weighted privacy budget of the data owner can be calculated as

$$\varepsilon_w = \sum_{i=1}^{m} \varepsilon_i w_i, \tag{13}$$

where $w_i = 1 - ((\varepsilon_i - \varepsilon_{\min})/\varepsilon_{\text{total}})$ and $\epsilon_{min} = \min\{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m\}$. It means that an ε_i will be multiplied by a smaller weight if ε_i is larger than $\varepsilon_{\min}$. Generally, the weighted privacy budget is smaller than ϵ_{total}, which indicates a PLDP scheme provides better privacy protection than an LDP scheme. The illustration of personalized privacy budget is shown in Figure 2.

4.3. Perturbation Mechanism for PLDP. In this section, we adapt the OUE to generate a perturbation mechanism for PLDP, which we name as personalized multiple optimized unary encoding (PMOUE), since there are multiple attributes in our scenario and the privacy budget can be personally allocated. We denote a data record with m unemptied attributes as $X = \{x_1, x_2, \ldots, x_m\}$, and the attribute x_i has l_i candidate values. According to OUE, the attribute $x_i = k$ is encoded to be a binary vector $v_i = (0, \ldots, 0, 1, 0, \ldots, 0)$ with the length l_i. The k-th bit in v_i is set to be 1, and the remaining bits are set to be 0. In order to protect the owner's data record, the bits in v_i are randomly flipped according to Algorithm 1, generating the perturbed element y_i. The parameters p and q in formula (6) depend on the privacy budget ϵ_i that are allocated to the element x_i. Finally, the data owner concatenates all of the y_i to be $Y = \{y_1, y_2, \ldots, y_m\}$ and sends Y to the server. The specific perturbation mechanism is described in Algorithm 1.

Theorem 4. *The perturbation mechanism PMOUE satisfies ϵ_{total}-PLDP, where $\epsilon_{total} = m \times \varepsilon_{average}$.*

Proof. For any inputs x', x'' with the same m unemptied attributes denoted by x_i' and x_i'', which are encoded to v_i' and v_i'', and the output y with m valid elements denoted by y_i, $i = 1, \ldots, m$, we have $\theta = ((\Pr(y|x')))/(\Pr(y|x'')) = \prod_{i=1}^{m} ((\Pr(y_i|x_i')))/(\Pr(y_i|x_i'))) = \prod_{i=1}^{m}\prod_{j=1}^{l_i} ((\Pr(y_i[j]|v_i'[j]))/(\Pr(y_i[j]|v_i''[j])))$.

According to formula (9), we have

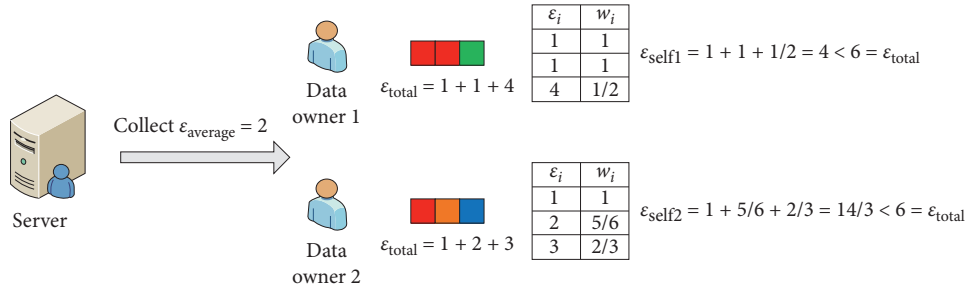

Figure 2: Illustration of personalized privacy budget.

$$\theta \le \prod_{i=1}^{m} \frac{\Pr(y_i[s_i] = 1 \mid v_i'[s_i] = 1)\Pr(y_i[t_i] = 0 \mid v_i'[t_i] = 0)}{\Pr(y_i[s_i] = 1 \mid v_i''[s_i] = 0)\Pr(y_i[t_i] = 0 \mid v_i''[t_i] = 1)}$$

$$= \prod_{i=1}^{m} \frac{p_i'(1 - q_i')}{q_i''(1 - p_i'')}$$

$$= \prod_{i=1}^{m} \frac{1 - q_i'}{q_i''}$$

$$= \prod_{i=1}^{m} \frac{e^{\varepsilon_i'}\left(e^{\varepsilon_i''} + 1\right)}{e^{\varepsilon_i'} + 1}$$

$$= e^{\sum_{i=1}^{m} \varepsilon_i'} \prod_{i=1}^{m} \frac{e^{\varepsilon_i''} + 1}{e^{\varepsilon_i'} + 1}$$

$$= e^{\varepsilon_{total}} \frac{\sum_{i=1}^{m} \sigma_i'' + 1}{\sum_{i=1}^{m} \sigma_i' + 1}, \tag{14}$$

where $\sum_{i=1}^{m} \sigma_i'$ and $\sum_{i=1}^{m} \sigma_i''$ are the m elementary symmetric polynomials about the variables $e^{\varepsilon_1'}, e^{\varepsilon_2'}, \cdots, e^{\varepsilon_m'}$ and $e^{\varepsilon_1''}, e^{\varepsilon_2''}, \cdots, e^{\varepsilon_m''}$ separately, i.e., $\sigma_1' = e^{\varepsilon_1'} + e^{\varepsilon_2'} + \cdots + e^{\varepsilon_m'}$, $\sigma_2' = e^{\varepsilon_1'}e^{\varepsilon_2'} + e^{\varepsilon_1'}e^{\varepsilon_3'} + \cdots + e^{\varepsilon_{m-1}'}e^{\varepsilon_m'}$, $\cdots$, $\sigma_m' = e^{\varepsilon_1'}e^{\varepsilon_2'}, \ldots, e^{\varepsilon_m'}$ [28].

Because $\lim_{x \longrightarrow 0} e^x = x + 1$, so $((\Pr(y \mid x))/(\Pr(y \mid x')))$

$$\le e^{\varepsilon_{total}} \frac{\sum_{i=1}^{m}(\varepsilon_i'' + 1) + \sum_{j=1}^{m}\sum_{i=1, i\neq j}^{m}\left(\varepsilon_i'' + \varepsilon_j'' + 1\right) + \cdots + \sum_{i=1}^{m}\varepsilon_i'' + 1 + 1}{\sum_{i=1}^{m}(\varepsilon_i' + 1) + \sum_{j=1}^{m}\sum_{i=1, i\neq j}^{m}\left(\varepsilon_i' + \varepsilon_j' + 1\right) + \cdots + \sum_{i=1}^{m}\varepsilon_i' + 1 + 1}$$

$$= e^{\varepsilon_{total}} \frac{\varepsilon_{total} + m + (m-1)\varepsilon_{total} + m + \cdots + \varepsilon_{total} + 1 + 1}{\varepsilon_{total} + m + (m-1)\varepsilon_{total} + m + \cdots + \varepsilon_{total} + 1 + 1}$$

$$= e^{\varepsilon_{total}}. \tag{15}$$

□

4.4. Estimation of the Joint Distribution. After receiving $Y = \{y_1, y_2, \ldots, y_m\}$ from each data owner, the server can estimate the joint distribution of multidimensional data. Here, we describe the estimation in two situations: (1) the frequency estimation for k-dimensional data that is included in some records, and (2) the frequency estimation for k-dimensional data that is not included in any records.

4.4.1. Situation 1. Since the k dimensions of data are included in some records, we group these records together for the frequency estimation. As presented in Section 3.3, the k dimensions of data are encoded to be the binary vector with the length $t = \sum_{i=1}^{k} l_i$, where l_i is the number of candidate values of the i-th attribute. Assuming there are s records that have these k dimensions, the frequency will be estimated from a binary matrix denoted as $\widehat{C}_{s\times t}$. Firstly, the server counts the number of '1' in each column, generating a vector $(\widehat{c}_1, \widehat{c}_2, \ldots, \widehat{c}_t)$. Then, to obtain an unbiased estimation, maximum likelihood estimation [29] is used to calibrate $\widehat{c}_i$ as follows:

$$c_i = \frac{\widehat{c}_i - Gq}{p - q} = \frac{\widehat{c}_i - \left(s/\left(e^{\varepsilon_{average}} + 1\right)\right)}{(1/2) - \left(s/\left(e^{\varepsilon_{average}} + 1\right)\right)}, \tag{16}$$

where c_i denotes the occurrence number of each candidate value, indicating the individual distribution of the attributes. G is the total number of records.

With the distribution of each dimension of data, the joint distribution of the k-dimensional data is estimated by LASSO regression. Firstly, the server encodes the candidate values of attributes to be binary string according to OUE. Next, the candidate values of different attributes are connected by Cartesian product and then transposed, resulting in the candidate matrix $M_{t\times r}$, where $t = \sum_{i=1}^{k} l_i$ is the bit length of the candidate value of the k-dimensional data and $r = \prod_{i=1}^{k} l_i$ is the total number of candidate values. Ensemble the candidate value frequencies of the k-dimensional data to be a vector $P = (p_1, p_2, \ldots, p_r)^T$, and we have

$$\mathrm{MP} = C, \tag{17}$$

where $C = (c_1, c_2, \ldots, c_t)^T$. Since the candidate value matrix $M_{t\times r}$ and the vector $C_{t\times 1}$ are both known, it seems that the unknown regression coefficient vector $P_{r\times 1}$ can be derived easily. In fact, the total number of joint candidate values r could be very large, but the occurrence frequencies of some candidate values are very small or even close to 0. That is to say, $P_{r\times 1}$ could be quite sparse.

As presented in Section 3.4, LASSO regression is quite suitable to solve the sparse linear regression. In this paper, the solution of frequency vector P in formula (17) is considered as a sparse linear regression problem where a shrunken vector is preferred. Thus, LASSO is used to obtain the frequency vector P. In Figure 3, we illustrate the whole process to estimate the frequency of the k-dimensional data in Situation 1.

(i) **Input:** $X = \{x_1, x_2, \ldots, x_m\}$: the raw data record with m unemptied attributes
ε_i: the privacy budget that allocated to attribute $x_i\,(i = 1, 2, \ldots, m)$
l_i: the number of the candidate values of each attribute $(i = 1, 2, \ldots, m)$
Output: $Y = \{y_1, y_2, \ldots, y_m\}$: the perturbed data record with m attributes

(1) Encoding:

$$v_i = \text{Encode}(x_i, l_i), i = 1, 2, \ldots, m$$

(2) Perturbation:

$$y_i = \begin{cases} \Pr(y_i[j] = 1 | v_i[j] = 1) = p_i = (1/2), \\ \Pr(y_i[j] = 0 | v_i[j] = 1) = 1 - p_i = (1/2), \\ \Pr(y_i[j] = 1 | v_i[j] = 0) = q_i = (1/(e^{\varepsilon} + 1)), \\ \Pr(y_i[j] = 0 | v_i[j] = 0) = 1 - q_i = (e^{\varepsilon}/(e^{\varepsilon} + 1)), \end{cases} \quad j = 1, 2, \ldots, l_i; i = 1, 2, \ldots, m,$$

(3) return $Y = [y_1, y_2, \ldots, y_m]$.

Algorithm 1: Personalized multiple optimized unary encoding (PMOUE).

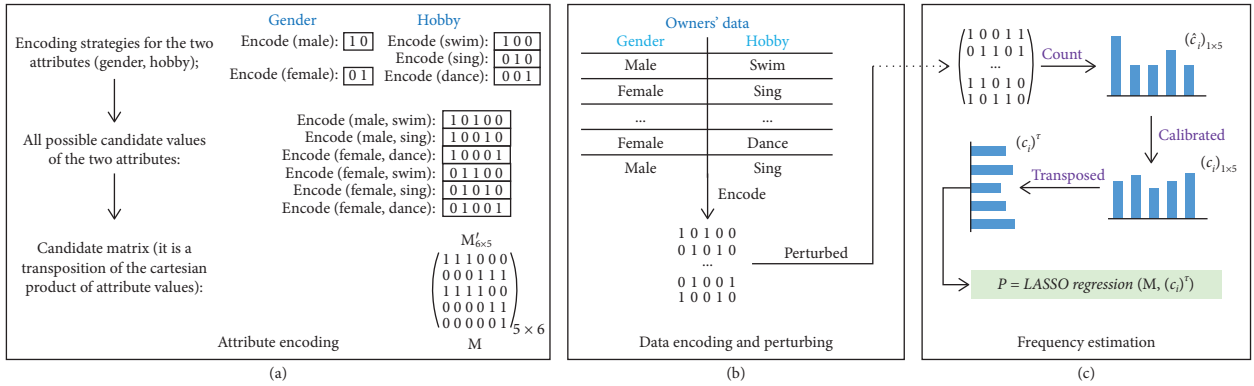

Figure 3: Illustration of LASSO scheme. (a) Attribute encoding. (b) Data encoding and perturbing. (c) Frequency estimation.

As shown in Figure 3, we consider an example with two attributes, i.e. gender and hobby, which, respectively, have two and three candidate values. Firstly, the candidate values are encoded to be the 2-bit and 3-bit binary vectors. The binary vectors of different candidate values are connected by Cartesian product, resulting in a matrix M' which then is transposed to be the candidate matrix M. Next, the data owners encode their data according to the encoding strategy and then perturb and upload the data to the server. In Situation 1, the considered k dimensions of data are included in some records. The server groups these records together to estimate the distribution of the frequency estimation. After receiving the perturbed data, the server counts the '1' bit in each column, resulting in a vector. The server calibrates the vector that indicates the frequencies of attribute values. Finally, the frequency vector P is calculated by LASSO regression algorithm with the M and C.

4.4.2. Situation 2. In this situation, the server wants to estimate the joint distribution of k attributes, denoted as $A = \{a_1, a_2, \ldots, a_k\}$, but there is no record containing all of these k attribute values. Accordingly, we divide the k attributes into two parts, i.e., A_1 and A_2, so that each part of the attributes is contained by some records. The A_1 and A_2 are considered as two multidimensional attributes with the domains Ω_1 and Ω_2, respectively. Then, we can estimate the distributions of A_1 and A_2 by LASSO regression algorithm as in Situation 1 separately and synthesize two distributions together.

The first step is to choose a division strategy and here we prefer the unbalanced one. That is to say, we prefer to let $\mathscr{A}_1$ have the most attributes in $\mathscr{A}$ and let $\mathscr{A}_2$ only have a small part of attributes. The second step is to allocate the attributes into $\mathscr{A}_1$ and $\mathscr{A}_2$. Intuitively, we hope $\mathscr{A}_1$ and $\mathscr{A}_2$ are as independent as possible. Then, the joint distribution of $\mathscr{A}_1$ and $\mathscr{A}_2$ can be calculated by direct multiplication. Information entropy [30] is a usual method to measure the amount of information and can also be used to identify the independency of the variable. Generally, a larger amount of information means larger independence. Based on the points above, we calculate the joint distribution of k attributes as described in Algorithm 2 and show its illustration in Figure 4.

5. Evaluation

In this section, we perform the experiments on four real datasets. The experimental results show the efficiency and superiority of our proposed notion PLDP in the aspect of accuracy and efficiency.

(i) **Input**: $\mathscr{A}$: the set of k attributes for which the sever wants to estimate the joint distribution and $\{X_i\}$ is the setof data records with $X_i = (x_{i1}, x_{i2}, \ldots, x_{id})$, where x_{ij} could be empty.
Output: HD: the joint distribution of $\mathscr{A}$
(1) **for** $r = 1$ to $(k/2)$ **do**
(2) Initialize a result list, $\text{list}_R = \{\varnothing\}$;
(3) Divided $\mathscr{A}$ into two parts: $\mathscr{A}_1$ and $\mathscr{A}_2$, where $|\mathscr{A}_1| = k - r$ and $|\mathscr{A}_2| = r$, and there are $n = \binom{k}{r}$ different divisions;
(4) **for** $i = 1$ to n **do**
(5) For the i-th division, if there are enough records that have all the attributes in $\mathscr{A}_1$ and $\mathscr{A}_2$, the sever estimatesthe joint distributions of $\mathscr{A}_1$ and $\mathscr{A}_2$ by LASSO regression algorithm, respectively. Denote the domains of candidate values of $\mathscr{A}_1$ and $\mathscr{A}_2$ as Ω_1 andΩ_2, respectively, and then the estimated joint distribution can be denoted as $P' = (p_1', p_2', \ldots, p_{|\Omega_1|}')$ and $P'' = (p_1'', p_2'', \ldots, p_{|\Omega_2|}'')$.
(6) Calculate $H' = \sum_{i=1}^{|\Omega_1|} p_i' \log p_i'$, $H'' = \sum_{i=1}^{|\Omega_2|} p_i'' \log p_i''$, and then the total information entropy in this division $H_i = H' + H''$.
(7) Finally, add the triplet $<P', P'', H_i>$ into list_R;
(8) **end for**
(9) **If** $\text{list}_R \neq \{\varnothing\}$ **Break**;
(10) **end for**
(11) Choose the triplet with the largest H_i and calculate the joint distribution with the corresponding P', P'' by the multiplication principle.

Algorithm 2: Information entropy-based multidimensional joint distribution estimation.

Candidate value	Probability
a1b1	0.25
a1b2	0.25
a2b1	0.25
a2b2	0.25

Candidate value	Probability
c1	0.47
c2	0.53

$H(\text{AB}) + H(\text{C}) = 0.60 + 0.30 = 0.90$

(a)

Candidate value	Probability
a1c1	0.30
a1c2	0.20
a2c1	0.20
a2c2	0.30

Candidate value	Probability
b1	0.70
b2	0.30

$H(\text{AC}) + H(\text{B}) = 0.59 + 0.26 = 0.85$

(b)

Candidate value	Probability
b1c1	0.70
b1c2	0.20
b2c1	0.05
b2c2	0.05

Candidate value	Probability
a1	0.65
a2	0.35

$H(\text{BC}) + H(\text{A}) = 0.38 + 0.28 = 0.64$

(c)

Figure 4: Illustration of information entropy-based scheme. (a) AB and C. (b) AC and B. (c) BC and A.

5.1. Experimental Setup

5.1.1. Environment. All the experiments were performed on a machine with Intel Core i5 CPU 2.50 GHz and 8 GB RAM, using Windows 10 and Python 3.5.2.

5.1.2. Datasets. We performed experiments on the following four real datasets, whose parameters are presented in Table 1. Abalone [31] contains the size of the abalone through physical measurements. Adult [31] is extracted by Barry Becker from the 1994 Census database. It contains personal information, such as "age," "relationship," and "money." Bank marketing [32] is related with direct marketing campaigns, which are based on phone calls, of a Portuguese banking institution. Car evaluation [31] contains several basic parameters of the car, which is derived from a simple hierarchical decision model. In order to simplify the experiments, we bucket the continuous and nonnumerical data into the discretized ones.

5.1.3. Evaluation Metrics. The performance of PLDP is measured by the accuracy of the distribution estimation and the time consumption during the estimation. The time consumption contains CPU time and IO cost. As in much previous work, we used the average variant distance (AVD) [33] to evaluate the estimation accuracy, which is defined as

$$\mathrm{AVD}(P, Q) = \frac{\sum_{\omega \in \Omega} |P(\omega) - Q(\omega)|}{2}, \tag{18}$$

where Ω is the domain of the multidimensional variable, $P(\omega)$ denotes the real joint distribution, and $Q(\omega)$ denotes the estimated distribution.

5.1.4. Setting of Privacy Budget. This paper considers the personalized privacy allocation. Here, we use a random function to assign the total privacy budget $\varepsilon_{\text{total}} = m \times \varepsilon_{\text{average}}$ randomly to the m valid elements for each data owner.

5.2. Comparison with LoPub on Adult Dataset. In LoPub [14], the data owners report their whole data with m attributes in their perturbed version. They perturb each dimension of data with a fixed privacy budget which is predefined by the server. After receiving the perturbed data, the server tried to estimate the joint distribution of k attributes $(1 \le k \le m)$ by using EM-based approach (LoPub-1), Lasso-based approach (LoPub-2), and a hybrid approach (LoPub-3). Please note that EM-based approach (LoPub-1) is not used in the case of $k = 1$ since the distribution of one attribute can be estimated directly by maximum likelihood estimation. In [14], the authors set a fixed privacy budget $\varepsilon = 2$ for each dimension of data, and each owner reports a 4-dimensional perturbed data. When the server estimates the joint distribution of k attributes, the total privacy budget equals to $\varepsilon \times k = 2k$. In our experiments, we test our scheme with three different settings to make a full comparison with LoPub [14].

5.2.1. Setting#1. Each data owner o_i reports an m_i-dimensional data perturbed with the total privacy budget $\varepsilon_{\text{total}} = m_i \times \varepsilon_{\text{average}}$, $\varepsilon_{\text{average}} = 2$, and $1 \le m_i \le 4$. Setting#1 is the normal setting of the proposed scheme, where the privacy budgets allocated on the attributes are uncertain. Accordingly, the joint distribution of $k = 1, 2, 3, 4$ attribute(s) could be estimated in both Situation 1 and Situation 2. The AVDs in Setting#1 are averaged from 10 estimations.

5.2.2. Setting#2. Each owner chooses two dimensions of his data randomly and then perturbed the data with a total privacy budget $\varepsilon_{\text{total}} = 2 \times \varepsilon_{\text{average}} = 2 \times 2 = 4$. Then, the joint distribution of $k = 1, 2, 3, 4$ attribute(s) is estimated. In Setting#2, the privacy budget allocated on the k attributes is equal to that in LoPub. In this way, the estimations of 1 and 2 attribute(s) follow the Situation 1 and the estimations of 3 and 4 attributes follow the Situation 2. The AVDs in Setting#2 are averaged from 10 estimations.

5.2.3. Setting#3. Here, each owner still chooses two dimensions of his data randomly to perturb and upload but sets different $\varepsilon_{\text{average}}$ when a different number of attributes are considered for distribution estimation. Specifically, the $\varepsilon_{\text{average}}$ is set to be k when the joint distribution of k attributes is estimated.

Figure 5(a) shows that our scheme achieves comparable accuracy to LoPub-3 despite that the privacy budgets are personally allocated to the attributes. Figure 5(b) shows that our scheme achieves better efficiency than LoPub, as our method needs not to iteratively scan the data owners' records. To sum up, the proposed scheme can provide personalized privacy allocation for each owner, which means a less weighted privacy budget. Accordingly, compared with LoPub, our scheme has higher security, lower time consumption, and comparable estimation accuracy. Note that, the results with regard to LoPub are directly taken from [14]. The test for LoPub-1 with $k = 4$ is not presented in [14] as it consumes too much time.

5.3. Results on Other Three Datasets. In this section, we test our scheme on four real-word datasets under Setting#1 and Setting#2 that are the same as that in Section 5.2. Figure 6 shows that, besides the dataset Adult, our scheme achieves similar accuracy on other three datasets. Our scheme allowed the data owner to only share a part of his data. Thus, the joint distribution of the high-dimensional data can only be obtained by synthesizing that of the low-dimensional data. We divide the attributes into two parts and decide the division by maximizing the sum of information entropies of the two parts. Finally, we synthesize the joint distribution by multiplication principle with the assumption that the two divided parts are independent of each other. This could decrease the estimation accuracy of our scheme.

Table 2 shows the experimental results of aggregating five-dimensional data. The settings of S#1 and S#2 are similar to the settings of Setting#1 and Setting#2. In S#1, each data owner o_i reports an m_i-dimensional data perturbed with the total privacy budget $\varepsilon_{\text{total}} = m_i \times \varepsilon_{\text{average}}$, $\varepsilon_{\text{average}} = 2$, and $1 \le m_i \le 5$. In S#2, each owner chooses three dimensions of

Table 1: Datasets.

Datasets	Attribute characteristics	#.Records (N)	#.Attributes (d)
Adult	Categorical, integer	48842	15
Abalone	Categorical, integer, real	4177	9
Bank marketing	Real	45211	17
Car evaluation	Categorical	1728	7

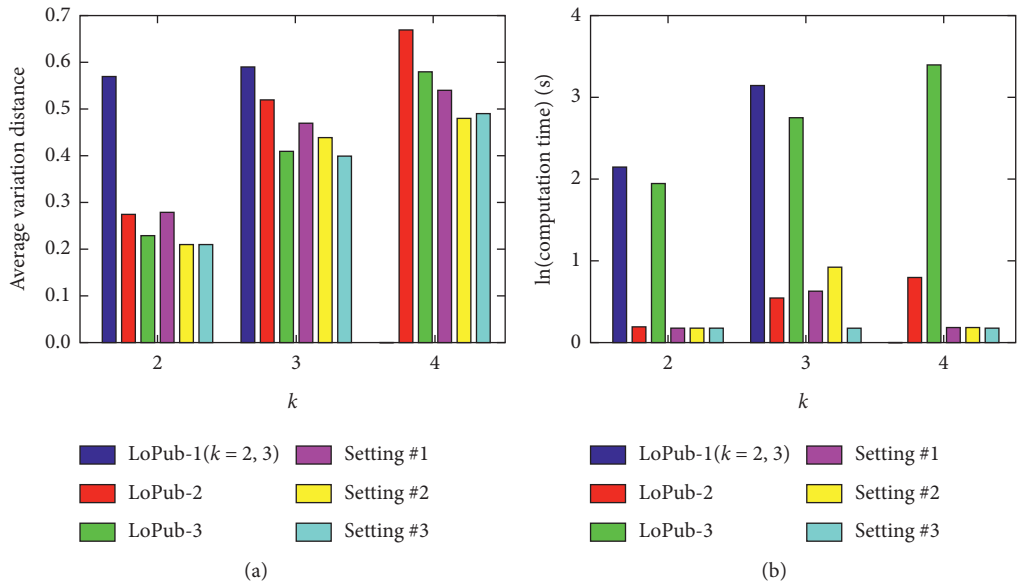

Figure 5: The comparison of the proposed scheme and LoPub on adult dataset. (a) AVD. (b) Computation time.

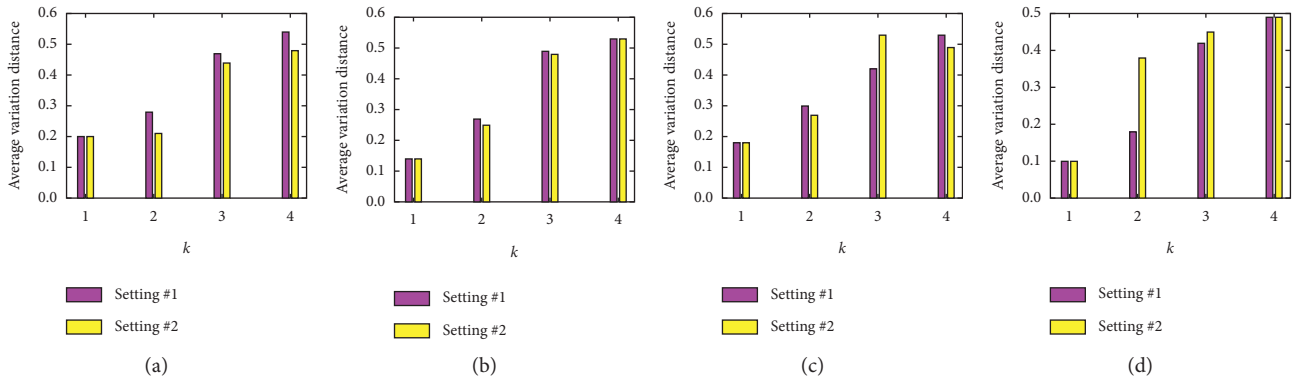

Figure 6: Accuracy vs. (k) ($\varepsilon_{\text{average}} = 2$). (a) Adult. (b) Abalone. (c) Bank marketing. (d) Car evaluation.

his data randomly and then perturbed the data with a total privacy budget $\varepsilon_{\text{total}} = 3 \times \varepsilon_{\text{average}} = 3 \times 2 = 6$. Then, the joint distribution of $k = 1, 2, 3, 4, 5$ attribute(s) is estimated.

Figure 7 compares the average computation time of two-dimensional data joint distribution estimation on the four real datasets with different average privacy budgets $\varepsilon_{\text{average}}$ and $k = 2$. As we can see, the computation takes only a few seconds. Moreover, when $\varepsilon_{\text{average}}$ is growing, the computation time increasing slowly even is unchanged. This is because the joint distribution is estimated by using LASSO regression, whose time complexity is mainly subject to the total number of records.

Table 2: Accuracy vs. k ($\varepsilon_{\text{average}} = 2$).

Datasets (settings)	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
Adult (S#1)	0.29	0.28	0.47	0.54	0.54
Adult (S#2)	0.29	0.27	0.43	0.48	0.51
Abalone (S#1)	0.36	0.36	0.42	0.53	0.53
Abalone (S#2)	0.36	0.36	0.39	0.45	0.49
Bank marketing (S#1)	0.36	0.36	0.42	0.48	0.53
Bank marketing (S#2)	0.36	0.36	0.41	0.43	0.45
Car evaluation (S#1)	0.27	0.34	0.40	0.49	0.52
Car evaluation (S#2)	0.27	0.34	0.37	0.41	0.47

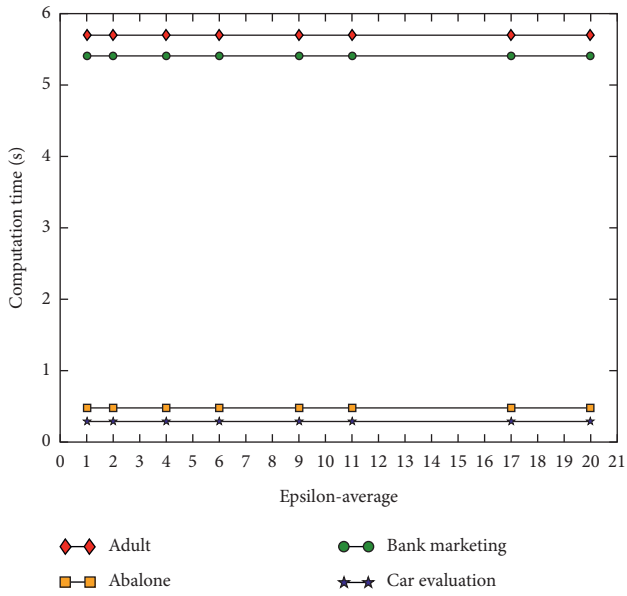

Figure 7: Computation time vs. $\varepsilon_{\text{average}}$ ($k = 2$).

6. Conclusion

In this paper, a new privacy notion PLDP is proposed to provide personalized privacy allocation under LDP. PLDP can be regarded as a generalized version of LDP. It allows users only to report their partial data and perturb them with distinct privacy budgets. Then, in order to estimate the joint distribution of multidimensional data, we develop an aggregation algorithm under PLDP, which is based on LASSO regression and information entropy. Finally, experiments on four real datasets validate the superiority and efficiency of PLDP, compared with the traditional LDP.

Data Availability

No data were used to support the findings of this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the Jiangsu Basic Research Programs-Natural Science Foundation under grant no. BK20181407, in part by the National Natural Science Foundation of China under grant nos. U1936118 and 61672294, in part by Six Peak Talent Project of Jiangsu Province (R2016L13), Qinglan Project of Jiangsu Province, and "333" Project of Jiangsu Province, in part by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) fund, in part by the Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET) fund, China. Zhihua Xia was supported by BK21+ program from the Ministry of Education of Korea.

References

[1] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, "What can we learn privately?" *SIAM Journal on Computing*, vol. 40, no. 3, pp. 793–826, 2011.

[2] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *Proceedings of the 2013 IEEE 54th annual Symposium on Foundations of computer science*, pp. 429–438, IEEE, Berkeley, CA, USA, October 2013.

[3] A. G. Thakurta, A. H. Vyrros, U. S. Vaishampayan et al., "Emoji frequency detection and deep link frequency," U.S. Patent-9-705908, 2017.

[4] B. Ding, J. Kulkarni, and S. Yekhanin, "Collecting telemetry data privately," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 3571–3580, Long Beach, CA, USA, December 2017.

[5] Q. Ye, H. Hu, X. Meng et al., "PrivKV: key-value data collection with local differential privacy," in *Proceedings of the2019 IEEE Symposium on Security and Privacy (SP)*, pp. 317–331, San Francisco, CA,USA, May 2019.

[6] T. Wang, N. Li, and S. Jha, "Locally differentially private frequent itemset mining," in *Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP)*, pp. 127–143, IEEE, San Francisco, CA,USA, May 2018.

[7] C. Dwork, "Differential Privacy," in *Proceedings of the33rd International Colloquium on Automata, Languages and Programming (ICALP)*, Venice, Italy, July 2006.

[8] Ú Erlingsson, V. Pihur, and A. Korolova, "Rappor: randomized aggregatable privacy-preserving ordinal response," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1054–1067, Scottsdale, AZ, USA, November 2014.

[9] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[10] R. Tibshirani, "Regression shrinkage and selection via the lasso: a retrospective," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 73, no. 3, pp. 273–282, 2011.

[11] J. W. Kim, D.-H. Kim, and B. Jang, "Application of local differential privacy to collection of indoor positioning data," *IEEE Access*, vol. 6, pp. 4276–4286, 2018.

[12] R. Bassily and A. Smith, "Local, private, efficient protocols for succinct histograms," in *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, pp. 127–135, Portland, OR, USA, June 2015.

[13] P. Kairouz, K. Bonawitz, and D. Ramage, "Discrete distribution estimation under local privacy," 2016, http://arxiv.org/abs/1602.07387.

[14] T. K. Ren, C.-M. Yu, W. Yu et al., "The expectation-maximization algorithm," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 9, pp. 2151–2166, 2018.

[15] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Processing Magazine*, vol. 13, no. 6, pp. 47–60, 1996.

[16] Z. Zhang, T. Wang, N. Li et al., "Calm:onsistent adaptive local marginal for marginal release under local differential privacy," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 212–229, Toronto, Canada, October 2018.

[17] S. Wang, L. Huang, M. Tian et al., "Personalized privacy-preserving data aggregation for histogram estimation," in *Proceedings of the2015 IEEE Global Communications*

Conference (GLOBECOM), pp. 1–6, Honolulu, HI, USA, November 2015.

[18] N. I. E. Yiwen, W. Yang, L. Huang et al., "A utility-optimized framework for personalized private histogram estimation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 4, pp. 655–669, 2018.

[19] R. Chen, H. Li, A. K. Qin et al., "Private spatial data aggregation in the local setting," in *Proceedings of the 2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pp. 289–300, IEEE, Helsinki, Finland, May 2016.

[20] R. Sarathy and K. Muralidhar, "Evaluating Laplace noise addition to satisfy differential privacy for numeric data," *Transactions on Data Privacy*, vol. 4, no. 1, pp. 1–17, 2011.

[21] N. McSherry, D. J. au, and O. au, "Privacy integrated queries: an extensible platform for privacy-preserving data analysis optimal differentially private mechanisms for randomised response," in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*, pp. 19–30, Providence, RI, USA, June 2009.

[22] N. Holohan, D. J. Leith, and O. Mason, "Optimal differentially private mechanisms for randomised response," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2726–2735, 2017.

[23] P. Kairouz, S. Oh, and P. Viswanath, "Extremal mechanisms for local differential privacy," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 2879–2887, Long Beach, CA, USA, January 2014.

[24] S. Wang, L. Huang, P. Wang et al., "Private weighted histogram aggregation in crowdsourcing," in *Proceedings of the International Conference on Wireless Algorithms, Systems, and Applications*, pp. 250–261, Yellow Mountains, China, August 2016.

[25] E. Yilmaz, M. Al-Rubaie, and J. M. Chang, "Locally differentially private naive bayes classification," 2019, http://arxiv.org/abs/1905.01039.

[26] T. Wang, J. Blocki, N. Li et al., "Locally differentially private protocols for frequency estimation," in *Proceedings of the 26th USENIX Security Symposium (USENIX Security 17)*, pp. 729–745, Vancouver, BC, Canada, August 2017.

[27] P. J. Li, T. Wang, M. Lopuhaä-Zwakenberg et al., "Estimating numerical distributions under local differential privacy," in *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pp. 621–635, Portland, OR, USA, June 2020.

[28] P. J. Schweitzer, "Elementary symmetric polynomials," *Siam Review*, vol. 9, no. 3, pp. 590-591, 1967.

[29] D.-Y. Pan, Y. Fang, and E. au, "Maximum likelihood estimation growth," *Curve Models and Statistical Diagnostics*, pp. 77–158, Springer, Berlin, Germany, 2002.

[30] D. Y. Tsai, Y. Lee, and E. Matsuyama, "Information entropy measure for evaluation of image quality," *Journal of Digital Imaging*, vol. 21, no. 3, pp. 338–347, 2008.

[31] S. Dua, P. Graff, and P. au, *UCI Machine Learning Repository*, University of California, School of Information and Computer Science., Irvine, CA, USA, 2019, http://archive.ics.uci.edu/ml.

[32] S. Moro, P. Cortez, and P. Rita, "A data-driven approach to predict the success of bank telemarketing," *Decision Support Systems*, vol. 62, pp. 22–31, 2014.

[33] R. Chen, Q. Xiao, Y. Zhang et al., "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*, pp. 129–138, Sydney, NSW, Australia, August 2015.