Pufferfish Privacy Mechanism Based on Multi-dimensional Markov Chain Model for Correlated Categorical Data Sequences

Zhicheng Xi(✉), Yingpeng Sang(✉), Hanrui Zhong, and Yongchun Zhang

Sun Yat-Sen University, Guangzhou, China
{xizhch,zhonghr3,zhangych65}@mail2.sysu.edu.cn,
sangyp@mail.sysu.edu.cn

Abstract. Differential privacy is a rigorous standard for protecting data privacy and has been extensively used in data publishing and data mining. However, because of its vulnerable assumption that tuples in the database are in-dependent, it cannot guarantee privacy if the data are correlated. Kifer et al. proposed the Pufferfish Privacy framework to protect correlated data privacy, while till now under this framework there is only some practical mechanism for protecting correlations among attributes of one individual sequence. In this paper, we extend this framework to the cases of multiple correlated sequences, in which we protect correlations among individual records, as well as correlations of attributes. Application scenarios can be different people's time-series data and the objective is to protect each individual's privacy while publishing useful information. We firstly define privacy based on Pufferfish privacy framework in our application, and when the data are correlated, the privacy level can be assessed through the framework. Then we present a multi-dimensional Markov Chain model, which can be used to accurately describe the structure of multi-dimensional data correlations. We also propose a mechanism to implement the privacy framework, and finally conduct experiments to demonstrate that our mechanism achieves both high utility and privacy.

Keywords: Pufferfish privacy · Multi-dimensional Markov Chain · Time series · Data correlations

1 Introduction

Big data era has come and it is called the "fourth paradigm" of scientific research. More and more databases are used in various fields such as healthcare, education, finance, population, transportation, science and technology, and have created huge social benefits. However, privacy concerns hinder the wildly use of these data. People would refuse to provide their sensitive information such as salary, diseases and user behavioral information. To this end, how to release useful information without revealing the individual's privacy has become a hot issue.

Dwork proposed the concept of differential privacy [2–5], which is still the state-of-the-art standard notion in data privacy. It provides a rigorous privacy guarantee that

H. Shen and Y. Sang (Eds.): PAAP 2019, CCIS 1163, pp. 430–439, 2020.
https://doi.org/10.1007/978-981-15-2767-8_38

it will not influence the outcome of any analysis when removing or adding a single database item. However, The initial framework of differential privacy is only effective for independent data records.

In practice, tuple correlation occurs naturally in datasets. User activity streams like time-series data, GPS trajectories and social networks typically generate records which are correlated. It has been shown that the data correlations can be utilized by attackers to improve their inferences about individuals and cause privacy leakage [7]. Group differential privacy has been proposed to solve this problem [5], which extends the privacy protection on individual to a group of correlated individuals. But the required noise may greatly destroy data utility.

Pufferfish was proposed by [8], which is based on differential privacy but can accommodate more situations. There are 3 important components in Pufferfish, a set of potential secrets S, a set of discriminative pairs S_{pairs}, and a set of data evolution scenarios $D(\theta \in D)$. It promises that the secret pairs are indistinguishable to the adversary. D captures how much knowledge the potential attackers have and then it can take the correlation of data into consideration. But the framework did not propose any specific perturbation algorithm to handle the correlation. Song et al. adopted the framework and used it to protect the privacy of time-series data such as physical activity measurements and power consumption data [11].

However, the prior work focuses on the correlations among individuals with only one attribute [14], or multiple attributes but only for one individual [12]. In this paper, we consider the correlations among individuals as well as the correlations among multiple attributes inside each sequence, such as different people's time-series data. These databases have wide applications, including stock markets, disease surveillance and real-time traffic monitoring. For example, in a database which records physical activities of members from the same family or company across time, there are different individual's records, and each record is a data sequence. Our goal is to publish aggregate statistics on individuals' activities without leaking the privacy of a specific individual, and here privacy is the activity at any given moment.

The contributions of our paper can be summarized as follows:

- We consider the simultaneous privacy protection for two types of correlations among categorical data sequences. One is the correlations among individuals, and the other is correlations inside each sequence.
- We propose a protection mechanism based on Pufferfish privacy by modelling correlations among variables employing the multi-dimensional Markov Chain.
- We conduct experiments on simulated data and demonstrate that our privacy mechanism provides both high privacy and utility guarantees.

2 Related Work

In the past decade, a growing body of work has been published on differential privacy [2–5]. As we explain earlier, differential privacy assumes that records are independent so it is not the right framework for the scenarios where records are correlated.

Correlated differential privacy has emerged to solve this problem. Kifer [7] was the first to raise the issue that differential privacy may not guarantee privacy without consideration of data correlations, and then proposed Pufferfish privacy [8], a generalization of differential privacy. It provides some specific instances of Pufferfish framework but is lack of specific mechanisms for many practical applications.

Existing privacy mechanisms for correlated data publishing can be classified into two types. The first one replaces the global sensitivity with new correlation-based parameters, such as dependence coefficient [9] and correlated sensitivity [15], and [10] used Maximal Information Coefficient to measure the correlations and achieved correlated differential privacy for big data publication. The other one uses appropriate models to describe the correlations between variables. [14] uses a modification of Pufferfish and proposed Bayesian differential privacy, which represents the data correlations by a Gaussian correlation model. Song proposed Markov Quilt Mechanism representing data correlation via a Bayesian Network [11]. There are also some efforts on time-series release such as [13] and high-dimensional data releasing based on Markov network [12]. However, these efforts only considered one-dimensional correlations of data. Therefore, they cannot be applied to simultaneously protect the two types of correlations. One type is the correlations among various sequences, and the other is the correlations inside each sequence.

3 Preliminaries

We will introduce some basic concepts in this section, including Pufferfish privacy mechanism, Multi-dimensional Markov Chain models, global sensitivity and Laplace mechanism. To start with, Table 1 lists notations and their explanations used across this paper.

3.1 Pufferfish Privacy Mechanism

We use Pufferfish framework as our privacy definition and extend it to apply in our cases. A Pufferfish framework consists of three parts, a set of potential secrets S, a set of discriminative pairs S_{pairs}, and a set of data evolution scenarios $D(\theta \in D)$. S captures what is protected, which is the set of secrets that refer to individual's private data. S_{pairs} captures how to protect, which means that the attackers cannot distinguish between the secret pairs. Finally, D captures how much knowledge the potential attackers have, which is a collection of plausible data generating distributions. In this paper, the correlations of data are controlled. Each $\theta \in D$ represents an adversary's belief about how to generate the data, and we should promise the indistinguishability.

Definition 3.1 (ϵ-Pufferfish(S, S_{pairs}, D) Privacy). Given set of potential secrets S, a set of discriminative pairs $S_{pairs}\big((s_i, s_j) \in S_{pairs}\big)$, a set of data evolution scenarios $D(\theta \in D)$, and a privacy parameter $\epsilon > 0$, M satisfies ϵ-Pufferfish(S, S_{pairs}, D) privacy if

$$P(M(X) = \omega | s_i, \theta) \leq e^{\epsilon} P(M(X) = \omega | s_j, \theta) \tag{1}$$

$$P(M(X) = \omega | s_j, \theta) \leq e^{\epsilon} P(M(X) = \omega | s_i, \theta) \tag{2}$$

Table 1. Table of notations

Symbol	Description
X	A database instance $\left\{x_n^k, k = 1, 2, \ldots, s\right\}$
$y_n^{(k)}$	The state probability distribution vector of the kth sequence at time n
S	Set of potential secrets
S_{pairs}	Discriminative pairs. $S_{pairs}\left(s_i, s_j\right) \subset S \times S$
D	The set of evolution scenarios: a conservative collection of plausible data generating distributions
M	A privacy mechanism over X
$P^{(jk)}$	The transition probabilities from the state of kth sequence at time n to the state of jth sequence at time $(n + 1)$
λ_{jk}	The weights between columns
F	A query function on X
GS_f	The global sensitivity of a query function on X
ϵ	The privacy budget

Equivalently,

$$e^{-\epsilon} \leq \frac{P(s_i|M(X) = \omega, \theta)}{P(s_j|M(X) = \omega, \theta)} \Big/ \frac{P(s_i|\theta)}{P(s_j|\theta)} \leq e^{\epsilon} \tag{3}$$

when s_i and s_j are such that $P(s_i|\theta) \neq 0$, $P(s_j|\theta) \neq 0$.

3.2 Multi-dimensional Markov Chain Models

Markov Chain models are widely used in the modeling of data sequences [1]. In our work, we use a multi-dimensional Markov Chain model for correlated data sequences such as sales demand data, stock index data and physical activities of a group individual. We assume that there are s sequences $\left\{y_n^{(k)}, k = 1, 2, \ldots, s\right\}$, and $y_n^{(k)}$ is the state probability distribution vector of the kth sequence at time n. Each sequence has m possible states in M. If the kth sequence is in state j with probability one at time n then we write $P\left\{y_n^{(k)} = j\right\} = 1$ or

$$y_n^{(k)} = \left(0, \ldots, 0, \underbrace{1}_{j}, 0, \ldots, 0\right)^T \tag{4}$$

The following conditions are satisfied in a multivariate Markov Chain model:

$$y_{n+1}^{(j)} = \sum\nolimits_{k=1}^{s} \lambda_{jk} P^{(jk)} y_n^{(k)}, (j = 1, 2, \ldots, s) \tag{5}$$

where $\sum_{k=1}^{s} \lambda_{jk} = 1, \lambda_{jk} \geq 0, 1 \leq j, k \leq s$. $P^{(jk)}$ are the transition probabilities from the state of kth sequence at time n to the state of jth sequence at time (n + 1), and λ_{jk} are the weights between columns.

The state probability distribution of the jth Chain at time (n + 1) is related to the state distribution of the s sequences at time n, but independent of the state before time n, which only hinges on the weighted average of $P^{(jk)} y_n^{(k)}$. The following is the matrix notation:

$$\begin{pmatrix} y_{n+1}^{(1)} \\ y_{n+1}^{(2)} \\ \vdots \\ y_{n+1}^{(s)} \end{pmatrix} = \begin{pmatrix} \lambda_{11} P^{(11)} & \lambda_{12} P^{(12)} & \cdots & \lambda_{1s} P^{(1s)} \\ \lambda_{21} P^{(21)} & \lambda_{22} P^{(22)} & \cdots & \lambda_{2s} P^{(2s)} \\ \vdots & \vdots & \vdots & \vdots \\ \lambda_{s1} P^{(s1)} & \lambda_{s2} P^{(s2)} & \cdots & \lambda_{ss} P^{(ss)} \end{pmatrix} \begin{pmatrix} y_n^{(1)} \\ y_n^{(2)} \\ \vdots \\ y_n^{(s)} \end{pmatrix} \tag{6}$$

Let $y_n = \left(y_n^{(1)}, y_n^{(2)}, \ldots, y_n^{(s)}\right)^T$, then $y_{n+1} = Q y_n$.

Lemma 1. For $1 \leq j, k \leq s$, if $\lambda_{jk} \geq 0$, then the matrix Q has a eigenvalue that is equal to 1, and the eigenvalues of Q are smaller than or equal to 1.

Lemma 2. For $1 \leq j, k \leq s$, assume that $\lambda_{jk} \geq 0$ and $P^{(jk)}$ is irreducible. Then there exists a stable vector $y = \left(y^{(1)}, y^{(2)}, \ldots, y^{(s)}\right)^T$ such that y = Qy and $\sum_{i=1}^{m} \left[y^{(j)}\right]_i = 1, 1 \leq j \leq s$.

In order to obtain the values of parameters, the transition probability matrix of each data sequence must be determined. Let $f_{i_j i_k}^{(jk)}$ represent the transition matrix from the state i_k in the sequence $\left\{y_n^{(k)}\right\}$ to the state i_j in the sequence $\left\{y_n^{(j)}\right\}$. Then the transition frequency matrix can be written as follows:

$$F^{(jk)} = \begin{pmatrix} f_{11}^{(jk)} & \cdots\cdots & f_{1m}^{(jk)} \\ f_{21}^{(jk)} & \cdots\cdots & f_{2m}^{(jk)} \\ \vdots & \vdots \quad \vdots & \vdots \\ f_{m1}^{(jk)} & \cdots\cdots & f_{mm}^{(jk)} \end{pmatrix} \tag{7}$$

And the following rule:

$$\hat{p}_{i_j i_k}^{(jk)} = \begin{cases} \dfrac{f_{i_j i_k}^{(jk)}}{\sum_{i_k=1}^{m} f_{i_j i_k}^{(jk)}}, & \sum_{i_k=1}^{m} f_{i_j i_k}^{(jk)} \neq 0 \\ 0, & in\ the\ other\ cases \end{cases} \tag{8}$$

Using this transition frequency matrix $F^{(jk)}$ and the normalized rule, one obtains the estimations of the matrix of transition probabilities $P^{(jk)}$:

$$\widehat{P}^{(jk)} = \begin{pmatrix} \hat{p}_{11}^{(jk)} & \cdots\cdots & \hat{p}_{1m}^{(jk)} \\ \hat{p}_{21}^{(jk)} & \cdots\cdots & \hat{p}_{2m}^{(jk)} \\ \vdots & \vdots \quad \vdots & \vdots \\ \hat{p}_{m1}^{(jk)} & \cdots\cdots & \hat{p}_{mm}^{(jk)} \end{pmatrix} \tag{9}$$

We also need to obtain the parameters λ_{jk}. There is a stable probability vector y in the multi-dimensional Markov Chain. We can estimate the vector y by calculating the probability of each state in each sequence, and is denoted as $\widehat{y} = \left(\widehat{y}^{(1)}, \widehat{y}^{(2)}, \ldots, \widehat{y}^{(s)}\right)^T$, then $\widehat{y} = Q\widehat{y}$. The values of λ_{jk} can be obtained by solving the following optimization problem:

$$\begin{cases} \min\limits_{\lambda} \max\limits_{i} \left| \left[\sum_{k=1}^{m} \lambda_{jk} \widehat{P}^{(jk)} \widehat{y}^{(k)} - \widehat{y}^{(j)} \right]_i \right| \\ subject\, to\; \sum_{k=1}^{s} \lambda_{jk} = 1,\; and\, \lambda_{jk} \geq 0,\; \forall k \end{cases} \tag{10}$$

This problem can be formulated as a linear programming problem. Let B be the condition-$B = \left[\widehat{P}^{(j1)} \widehat{y}^{(1)} \middle| \widehat{P}^{(j2)} \widehat{y}^{(2)} \middle| \ldots | \widehat{P}^{(js)} \widehat{y}^{(s)} \right]$, the model can be written as follows. For each j:

$$\min_{\lambda} w_j$$

Subject to

$$\begin{cases} \begin{pmatrix} w_j \\ w_j \\ \vdots \\ w_j \end{pmatrix} \geq \widehat{y}^{(j)} - B \begin{pmatrix} \lambda_{j1} \\ \lambda_{j2} \\ \vdots \\ \lambda_{js} \end{pmatrix}, \\ \begin{pmatrix} w_j \\ w_j \\ \vdots \\ w_j \end{pmatrix} \geq -\widehat{y}^{(j)} + B \begin{pmatrix} \lambda_{j1} \\ \lambda_{j2} \\ \vdots \\ \lambda_{js} \end{pmatrix}, \\ w_j \geq 0, \\ \sum_{k=1}^{s} \lambda_{jk} = 1,\; and\, \lambda_{jk} \geq 0,\; \forall j \end{cases} \tag{11}$$

3.3 Additional Notion

We introduce some additional definitions and notation to conclude this section.

Definition 3.4 (global sensitivity). Let f be a function that maps a dataset into a fixed-size vector of real numbers (i.e.$X \to R^d$). For any two neighboring databases X and X', the sensitivity of f is defined as

$$GS_f = \max_{X,X'} \left\| f(X) - f\left(X'\right) \right\|_p \tag{12}$$

Where p denotes L_p norm used to measure Δf, and we usually use L_1 norm.

For any query function *F:* $X \to R^d$, the privacy mechanism M

$$M(X) = f(X) + Z \tag{13}$$

Satisfies ϵ-differential privacy, where $Z \sim Lap(\Delta f/\epsilon)$. We use $Lap(\sigma)$ to denote a Laplace distribution with mean 0 and scale parameter σ. Recall that this distribution satisfies the density function: $h(x) = \frac{1}{2\sigma} e^{-|x|/\sigma}$.

4 A Mechanism for 2-Dimensional Correlated Data

4.1 Problem Statement

We consider a more restricted setting when the database X are several categorical data sequences. We assume that there are s categorical sequences and each has m possible states in M. Their dependence can be described by multi-dimensional Markov Chain model, and the goal is to keep the value of each x_i^k private. We next use two examples to illustrate the problem.

Example 1: A Group Physical Activity Measurement. A is the set of activities such as {walking, sleeping, working} and s_t^{k*a} denotes the event that the kth person's state is activity a at moment t, i.e., $x_t^k = a$. In the Pufferfish framework, we set S as $\{s_t^{k*a} : k = 1, \ldots, s, t = 1, \ldots, T, a \in A\}$, so the activity at any specific moment t of each person is a secret. S_{pairs} is the set of all pairs (s_t^{k*a}, s_t^{k*b}) for a, b in A and for all t and each person; in other words, for all pairs a and b, the attackers cannot tell whether this person is doing activity a or activity b at any time. D is a set of possible distributions to generate the data, which captures how people switch between activities and how people influence each other. A plausible belief is to set D be a set of multi-dimensional Markov Chains where each state is an activity in A. Each multi-dimensional Markov Chain can be represented by an initial distribution y_1 which represents the initial state of each sequence, the transition probabilities $P^{(jk)}$ and the weights between columns λ_{jk}. For example, we have two activities {walking, working} and use $(1, 0)^T$ to represent walking. There are two sequences in the dataset. Thus, a distribution $\theta \in D$ is represent by a tuple

$$\left\{ y_1, \begin{bmatrix} P^{11} & P^{12} \\ P^{21} & P^{22} \end{bmatrix}, \begin{bmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{bmatrix} \right\}$$

Then such D can be the set:

$$\left\{ \begin{matrix} \left(\begin{bmatrix} (0,1)^T \\ (1,0)^T \end{bmatrix}, \begin{bmatrix} \begin{bmatrix} 1 & 0.5 \\ 0 & 0.5 \end{bmatrix} & \begin{bmatrix} 0.7 & 0.6 \\ 0.3 & 0.4 \end{bmatrix} \\ \begin{bmatrix} 0.8 & 0.5 \\ 0.2 & 0.5 \end{bmatrix} & \begin{bmatrix} 0.9 & 0.6 \\ 0.1 & 0.4 \end{bmatrix} \end{bmatrix}, \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} \right), \\ \left(\begin{bmatrix} (1,0)^T \\ (0,1)^T \end{bmatrix}, \begin{bmatrix} \begin{bmatrix} 0 & 0.5 \\ 1 & 0.5 \end{bmatrix} & \begin{bmatrix} 0.5 & 0.5 \\ 0.6 & 0.4 \end{bmatrix} \\ \begin{bmatrix} 0.4 & 0.3 \\ 0.6 & 0.7 \end{bmatrix} & \begin{bmatrix} 0.9 & 0.6 \\ 0.1 & 0.4 \end{bmatrix} \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \end{matrix} \right\}$$

Example 2: Sales Demand Data Sequences. The database consists of a soft-drink company's sales demand data. The company has 5 products {A, B, C, D, E} and each product is labeled as its moving rate of sales volume - {very fast-moving, fast-moving, standard, slow-moving, very slow-moving, no sales volume}. Each customer of the company has 5 sales demand data sequences. We can use the database to reduce the company's inventory and maximize the needs of each customer, but we cannot reveal customer's privacy which means the adversary cannot infer the customer's demand for all products at a specific time. Let M be the moving states set and let s_t^{k*m} denote the event that the kth product's state is m at time t, namely, $x_t^k = m$. In the Pufferfish framework, we set S as $\{s_t^{k*m} : k = 1, \ldots, 5, t = 1, \ldots, T, m \in M\}$, so the state at each time t of each product is a secret. S_{pairs} is the set of all pairs (s_t^{k*m}, s_t^{k*n}) for m, n in M and for all t and each product. Similarly, D can also be a set of multi-dimensional Markov Chains.

4.2 Our Mechanism

In our mechanism, we first use multi-dimensional Markov Chains to describe the 2-dimensional correlation and get the set of all possible distributions which can generate the data. Then we adopt the Pufferfish framework and customize our privacy definition for our application. At last, we use the concept of interpretation by adding appropriate noise to the result and then achieve both utility and privacy.

Our mechanism is based on the Laplace mechanism in differential privacy which adds noise to the result of F proportional to the global sensitivity. In our mechanism, we use the worst-case distance between the distribution $P(F(X)|s_i, \theta)$ and $P\big(F(X)|s_j, \theta\big)$ for a secret pair (s_i, s_j). First, we use the idea of Earth Mover's Distance (EMD) to represent two probability distributions' distance.

Definition 4.1. Let μ, ν be two probability distributions on R, and let $\Gamma(\mu, \nu)$ be the set of all joint distributions. The distance between μ and ν is defined as:

$$Distance_{\infty}(\mu, \nu) = inf_{\gamma \in \Gamma(\mu,\nu)} \max_{(a,b) \in support(\gamma)} |a - b| \tag{14}$$

The Earth mover's distance is the minimum shift probability mass between μ and ν which in our mechanism is $P(F(X)|s_i, \theta)$ and $P\big(F(X)|s_j, \theta\big)$. To guarantee the Pufferfish privacy, we add Laplace noise to the result of the query F proportional to the $Distance_{\infty}\big(P(F(X)|s_i, \theta), P\big(F(X)|s_j, \theta\big)\big)$. We describe the full mechanism in Algorithm 1.

```
Algorithm 1 : A Mechanism for 2-dimensional Correlated Data
Given Database $X$, query $F$, Pufferfish framework($S$, $S_{pairs}$, $D$), privacy parameter $\epsilon$
for all $(s_i, s_j) \in S_{pairs}$ and all $\theta \in D$ such that $P(s_i|\theta) \neq 0$ and $P(s_j|\theta) \neq 0$ do
    Set $\mu_{i,\theta} = P(F(X) =\cdot |s_i, \theta)$, $\mu_{j,\theta} = P\big(F(X) =\cdot \,|s_j, \theta\big)$.
    Calculate $Distance_{\infty}(\mu_{i,\theta}, \mu_{j,\theta})$
end for
Set $Distance = sup_{(s_i,s_j) \in S_{pairs}, \theta \in D} Distance_{\infty}(\mu_{i,\theta}, \mu_{j,\theta})$.
return $F(X) + Z$, where $Z \sim Lap(\frac{Distance}{\epsilon})$
```

For given Database X, query F, Pufferfish framework (S, S_{pairs}, D), and privacy parameter ϵ, we find the supremum of the distance (EMD) between $\mu_{i,\theta}$ and $\mu_{j,\theta}$ through all S_{pairs} and D. Then, we add the Laplace noise to the result of F proportional to the distance we find. The mechanism for 2-dimensional correlated data satisfies the pufferfish privacy.

5 Experiments

We apply our mechanism to the simulated data which is generated by a multi-dimensional Markov Chain of two sequences ($s = 2$) and each sequence with length $T = 100$ and states $\{0, 1\}$. We employ this prototype simulation in order to achieve an efficient implementation of our algorithm.

First, we generate the database X which is determined by initial distribution for two sequences with two parameters $q_0^1 = P\left(X_1^1 = 0\right)$ and $q_0^1 = P\left(X_1^2 = 0\right)$, the transition probabilities $P^{(jk)}$ and the weights between columns λ_{jk} which are equal to 0.5 in our setting. The transition probabilities are determined by four transition matrices and each matrix such as $P^{(11)}$, $P^{(12)}$, $P^{(21)}$, *or* $P^{(22)}$ is determined by parameters p_0^{jk} and p_1^{jk}, in which $p_0^{jk} = P(X_{i+1}^j = 0|X_i^k = 0)$ and $p_1^{jk} = P(X_{i+1}^j = 1|X_i^k = 1)$.

The query $F(X) = \frac{1}{T*s}\sum_{k=1}^{s}\sum_{i=1}^{T} X_i^{\mathrm{k}}$. Then we calculate the conditional probability $P(F(X) = \cdot|s_i, \theta)$ and $P\left(F(X) = \cdot|s_j, \theta\right)$ and measure the distance between them by Earth Mover's Distance. The privacy budget ϵ varies in $\{0.2, 0.5, 1, 2, 5\}$. We compare the actual *F(X)* with our output result and show the average L_1 error between them. We use group differential privacy as our baseline which assumes that all variables are correlated and adds $Lap(1/\epsilon)$ noise to each bin. Table 2 shows the result of our experiments.

Table 2. L_1 error of frequency of state 1

ϵ	0.2	0.5	1	2	5
Our mechanism	3.1498	1.4735	0.4326	0.1252	0.0243
Group DP	4.3157	2.3584	0.6324	0.1432	0.1025

From Table 2, we can see that our mechanism is more accurate than group differential privacy. As expected, the L_1 error decreases as the private budget ϵ increases which shows that smaller ϵ means more privacy. The experiments show that our mechanism achieves both higher utility and privacy than group differential privacy.

6 Conclusion

We propose a Pufferfish privacy mechanism for correlated categorical data sequences, such as a group of physical activity measurements, sales demand data sequences, and other time-series datasets. We use the multi-dimensional Markov Chain model to represent the correlations among individuals and inside each sequence. Experiments with simulated data show that our mechanism achieves both high utility and privacy.

There are still some aspects for our work to be improved in the future. The computational efficiency can be improved by exploiting structural information of multi-dimensional Markov Chains. Experiments also need to be conducted on real-world datasets. Some other types of correlated data, such as semi-structured data, graph data and large-scale data, also requires novel models and privacy mechanisms.

Acknowledgements. This work was supported by the National Key Research and Development Program of China (No. 2017YFB0203201), the Science and Technology Program of Guangdong Province, China (No. 2017A010101039), and the Science and Technology Program of Guangzhou, China (No. 201904010209).

References

1. Ching, W., Zhang, S., Ng, M.: On multi-dimensional Markov chain models. Pac. J. Optim. **3**(2), 235–243 (2007)
2. Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., Naor, M.: Our data, ourselves: privacy via distributed noise generation. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 486–503. Springer, Heidelberg (2006). https://doi.org/10.1007/11761679_29
3. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_14
4. Dwork, C.: Differential privacy: a survey of results. In: Agrawal, M., Du, D., Duan, Z., Li, A. (eds.) TAMC 2008. LNCS, vol. 4978, pp. 1–19. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-79228-4_1
5. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Found. Trends® Theor. Comput. Sci. **9**(3–4), 211–407 (2014)
6. Humbert, M., Trubert, B., Huguenin, K.: A Survey on Interdependent Privacy (2019)
7. Kifer, D., Machanavajjhala, A.: No free lunch in data privacy. In: Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data, pp. 193–204. ACM (2011)
8. Kifer, D., Machanavajjhala, A.: Pufferfish: a framework for mathematical privacy definitions. ACM Trans. Database Syst. (TODS) **39**(1), 3 (2014)
9. Liu, C., Chakraborty, S., Mittal, P.: Dependence makes you vulnberable: differential privacy under dependent tuples. In: NDSS, vol. 16, pp. 21–24 (2016)
10. Lv, D., Zhu, S.: Achieving correlated differential privacy of big data publication. Comput. Secur. **82**, 184–195 (2019)
11. Song, S., Wang, Y., Chaudhuri, K.: Pufferfish privacy mechanisms for correlated data. In: Proceedings of the 2017 ACM International Conference on Management of Data, pp. 1291–1306. ACM (2017)
12. Wei, F., Zhang, W., Chen, Y., Zhao, J.: Differentially private high-dimensional data publication via Markov network. In: Beyah, R., Chang, B., Li, Y., Zhu, S. (eds.) SecureComm 2018. LNICST, vol. 254, pp. 133–148. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01701-9_8
13. Wang, H., Xu, Z.: CTS-DP: publishing correlated time-series data via differential privacy. Knowl.-Based Syst. **122**, 167–179 (2017)
14. Yang, B., Sato, I., Nakagawa, H.: Bayesian differential privacy on correlated data. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, pp. 747–762. ACM (2015)
15. Zhu, T., Xiong, P., Li, G., et al.: Correlated differential privacy: hiding information in non-IID data set. IEEE Trans. Inf. Forensics Secur. **10**(2), 229–242 (2014)