On the Lower Bounds for Learning and Testing Markov Chains

Haitong Liu *
antonyhtliu@link.cuhk.edu.hk

Xiwei Cheng *
xwcheng@link.cuhk.edu.hk

Abstract

We study the problem of learning and testing non-i.i.d. distributions, especially Markov chains (which is regarded as one of the simplest non-i.i.d. distributions). Observing a trajectory from an irreducible Markov chain, the goal of learning is to estimate its underlying transition matrix, and the goal of testing is to decide whether the trajectory is drawn from a given reference transition matrix. We have two main results, both focusing on the impossibility perspective.

Our first contribution proves a lower bound on the sample complexity of any learner for Markov chains. The lower bound matches the algorithm proposed by [1] up to constant factors, thus proving the optimality of its algorithm and solving one open problem therein.

Our second contribution considers one additional setting: dependent differential privacy (DDP), which is a scheme that protects sensitive data against attackers with partial information in statistical estimation tasks. We prove that no asymptotic privacy guarantee in the DDP sense is achievable for any practical testing algorithm for general Markov chains.

Keywords: Markov chains, learning and testing, lower bound

*This work is a group report for ESTR4998 Graduation Thesis, supervised by Prof. Siu On Chan. The two authors contribute equally to this work, and the author list is ordered alphabetically. More precisely, Sections 2.1, 2.3, 5, 6 are mainly contributed by Haitong Liu; Sections 2.2, 4 are mainly contributed by Xiwei Cheng; Sections 1, 3 are joint work by the two authors.

1 Introduction

Learning and testing discrete distributions is an active research area (see e.g. [2–6]). While learning and testing from independent and identically distributed (i.i.d.) samples have been well-studied [2, 7], the same problem on samples drawn from a dependent distribution remains largely unexplored.

We mainly focus on learning and testing the transition matrix M of an irreducible Markov chain from its trajectory, as considered in [1, 8–10]. Consider a finite irreducible Markov chain over states $[n]$ with transition matrix M. Given the initial state X_0 (or more generally, an initial distribution), a Markovian trajectory $X_1, X_2, \dots$ can be generated according to the current state and its transition probabilities, i.e., $\Pr(X_{t+1} = j | X_t = i) = M_{ij}$ for all $t \geq 0$. With the observation of a single Markovian trajectory, we are interested in giving an estimation $\hat{M}$ of its underlying transition matrix M. Following [8], the quality of the estimator $\hat{M}$ is measured by its closeness to M under the infinity matrix norm.

We first introduce the state-of-the-art algorithm (to the best of our knowledge) for learning a Markov chain under infinity matrix norm, proposed in [1]. The algorithm leverages the memoryless property of Markov chains and designs a "state-wise" learning scheme by invoking learners for discrete distributions. More precisely, although the Markovian trajectory follows a dependent distribution, one could prove that, in an infinitely long trajectory, the succeeding states of a certain fixed state (say state i) are distributed according to i.i.d. samples from M_i. Then, [1] introduces the notion of k-cover time, i.e., the expected minimal length of a random walk to cover every state at least k times. Hence, the algorithm could estimate the Markov chain when every state has been visited enough number of times, upon invoking discrete distribution learners to learn the outgoing transitions of every state.

Our first contribution is that we prove a tight lower bound on the sample complexity of any learner for Markov chains, which matches the sample complexity of the algorithm in [1] up to constant factors, thus solving an open problem raised in [1].

The other part of our work focuses on testing Markov chains in the differential privacy context. We find that the (pure) differential privacy for such tasks almost comes for free, and we consider a strictly stronger definition, known as dependent differential privacy (DDP), as proposed in both [11] and [12]. And our second contribution is to prove a constant lower bound for the privacy parameter of DDP for testing Markov chains, which also indicates that such privacy guarantees are beyond reach when considering learning and testing tasks.

1.1 Organizations

In section 2 we discuss the basic notions of distributional learning and testing on discrete distributions and Markov chains, and the definition of dependent differential privacy. In Section 3 we state the main results of this work. We prove a lower bound for learning irreducible Markov chains in Section 4, and a lower bound for dependent differential private testing on Markov chains is given in Section 5. Section 6 discusses the conclusion and some potential follow-up directions.

As this is a final year project report, it is worth mentioning that all the contents in Sections 4 and 5, if not specified or referenced explicitly, are the original work by the two authors.

1.2 Notations

Throughout this report, we use $[n]$ to denote the discrete finite set $\{1, 2, 3, \ldots, n\}$. We let Δ^{n-1} be the $(n-1)$-dimensional probability simplex. We define the following measurements for the distance between discrete probability measure $p(\cdot)$ and $q(\cdot)$: $d_{\mathrm{TV}}\left(p, q\right) = \max_{A \subset [n]} |p\left(A\right) - q\left(A\right)|$ is the total variation distance, where $p(A) = \sum_{i \in A} p(i)$, and $d^2_{\mathrm{H}}\left(p, q\right) = \frac{1}{2}\sum_{i=1}^{n}\left(\sqrt{p\left(i\right)} - \sqrt{q\left(i\right)}\right)^2$ is the Hellinger distance. And we formally define matrix infinity norm as

$$||M - M'||_\infty = \max_{i \in [n]} \sum_{j \in [n]} |M_{ij} - M'_{ij}| = \max_{i \in [n]} 2d_{\mathrm{TV}}(M_i, M'_i). \tag{1}$$

Let $X^m = \{X_1, \cdots, X_m\}$ be a sequence of samples drawn from a fixed distribution over $[n]^m = [n] \times [n] \times \cdots \times [n]$, where the product is the standard Cartesian product. We also denote the number of occurrences of i in X^m as $N_i(X^m)$. Given two independent random variables X, Y from distributions p, q respectively, we denote the (joint) distribution of (X, Y) as $p \otimes q$, known as the product distribution of p and q. Moreover, we introduce the following definition of (ϵ, δ)-learner:

Definition 1 ((ϵ, δ)**–Learner**) *Consider an algorithm $\mathcal{A}$ that takes as input i.i.d. samples (respectively a Markovian trajectory) X^m from a discrete* *distribution (resp. a Markov chain) M over n states and outputs an estimation $\hat{M} \triangleq \mathcal{A}(X^m, n)$ of the distribution. Given constants $\epsilon, \delta \in (0, 1)$, $\mathcal{A}$ is called (ϵ, δ)–Learner for discrete distributions (resp. Markov chains) with sample complexity $k(n, \epsilon, \delta)$, if $d_{TV}(\hat{D}, D) < \epsilon$ (resp. $||\hat{M} - M||_\infty < \epsilon$) with probability $\geq 1 - \delta$.*

We make use of standard Bachmann-Landau asymptotic notation $\mathcal{O}(\cdot)$, $o(\cdot)$, $\Theta(\cdot)$ and $\Omega(\cdot)$. We write the floor and ceil functions as $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ respectively.

Remark 1 *As we are mainly interested in asymptotic bounds, we implicitly require ϵ and δ, and a few auxiliary hyperparameters to be sufficiently small, as specified in the rest of this work. Note that such consideration would not hurt the asymptotic bounds.*

2 Preliminaries

2.1 Learning Discrete Distributions

We will first introduce a standard algorithm for learning (finite) discrete distributions, then prove that this algorithm is optimal in terms of sample complexity up to constant factors. However, the emphasis will be on the second part, namely, the lower bound results. Our general reference for this section is [13].

2.1.1 Upper Bound for Learning Discrete Distributions

Theorem 1 *There exists an algorithm that (ϵ, δ)-learns discrete distributions on $[n]$ with at most $m = \mathcal{O}\left(\frac{n+\log 1/\delta}{\epsilon^2}\right)$ samples.*

Proof of theorem 1: The upper bound for learning is straightforward, and can be achieved by the empirical estimator. One way to analyze its performance relies on the following well-known concentration bound[1]:

Fact 1 (**Hoeffding's bound**) *Let $X_1, X_2, \ldots, X_m$ be independent random variables such that $\forall i \in [m]$, $\mathbb{E}[X_i] = \mu$ and there exist constants $a < b$, $\Pr(a \leq X_i \leq b) = 1$, then*

$$\Pr\left(\left|\frac{1}{m}\sum_{i=1}^{m} X_i - \mu\right| \geq \epsilon\right) \leq 2\exp\left(-\frac{2m\epsilon^2}{(b-a)^2}\right).$$

Recall that $d_{\mathrm{TV}}(p, q) = \max_{S \subseteq [n]} p(S) - q(S)$ for probability measure p, q, so it suffices to prove $\Pr(\hat{p}(S) - p(S) \geq \epsilon) \leq \frac{\delta}{2^n}$ $\forall S \subseteq [n]$ for some $m = \mathcal{O}((n + \log 1/\delta)/\epsilon^2)$ and then apply the union bound.

Now let $Y_i = \mathbb{1}_{\{X_i \in S\}}$ be the indicator of whether X_i is in subset S, it is obvious that $\mathbb{E}[Y_i] = p(S)$ and $0 \leq Y_i \leq 1$. If we take $m = \frac{1}{2\epsilon^2}\log\frac{2^{n+1}}{\delta} =$

[1] To see a proof of Hoeffding's bound, one can refer to e.g. Chapter 4 in [14].

$\frac{n+1+\log 1/\delta}{2\epsilon^2} = \mathcal{O}\left(\frac{n+\log 1/\delta}{\epsilon^2}\right)$, applying Hoeffding's bound (Fact 1), we obtain

$$\begin{aligned}
\Pr\left(\hat{p}(S) - p(S) \geq \epsilon\right) &= \Pr\left(\sum_{i=1}^{m} \frac{Y_i}{m} - p(S) \geq \epsilon\right) \\
&= \Pr\left(\frac{1}{m}\sum_{i=1}^{m} Y_i - \mathbb{E}\left[Y_i\right] \geq \epsilon\right) \\
&\leq 2\exp\left(-2m\epsilon^2\right) \\
&= 2 \cdot \frac{\delta}{2^{n+1}} = \frac{\delta}{2^n}.
\end{aligned} \tag{2}$$

There are at most $2^n - 1$ such non-empty $S \subsetneq [n]$, then by union bound, we have

$$\Pr\left(d_{\mathrm{TV}}(\hat{p}, p) \geq \epsilon\right) = \Pr\left(\exists S,\ \hat{p}(S) - p(S) \geq \epsilon\right) < 2^n \Pr\left(\hat{p}\left(S\right) - p\left(S\right) \geq \epsilon\right) = \delta. \quad \blacksquare$$

2.1.2 Distinguishing Two Distributions

For the lower bound, the following lemma serves as one useful building block for further hardness results in the remaining sections:

Lemma 1 (Distinguishing two distributions)
Distinguishing discrete probability distributions p, q with success probability $\geq 1 - \delta$ requires at least $m = \Omega\left(\frac{\log 1/\delta}{d_H^2(p,q)}\right)$ samples.

Proof of Lemma 1: The proof relies on the following facts:

Fact 2 *If $d_{TV}(p, q) < 1 - 2\delta$, no algorithm can distinguish between p and q with success probability $\geq 1 - \delta$ from a single sample.*

Fact 3 (**Properties of Hellinger distance**)
$\left(1 - d_H^2\left(p, q\right)\right)^2 \leq 1 - d_{TV}^2\left(p, q\right), d_H^2\left(p^{\otimes m}, q^{\otimes m}\right) = 1 - \left(1 - d_H^2\left(p, q\right)\right)^m.$

To see Fact 2, consider $X \sim \theta$, where $\theta \sim$ Uniform$\{p, q\}$, and any randomized algorithm can be characterized by conditional probability distribution $P_{\hat{\theta}|X}$.

Then we have

$$\begin{aligned}\Pr(Success) &= \frac{1}{2}\Pr(Success|\theta = p) + \frac{1}{2}\Pr(Success|\theta = q) \\ &= \frac{1}{2}\sum_x p(x)P_{\hat{\theta}|X}(p|x) + \frac{1}{2}\sum_x q(x)P_{\hat{\theta}|X}(q|x) \\ &= \frac{1}{2}\sum_x p(x)P_{\hat{\theta}|X}(p|x) + \frac{1}{2}\sum_x q(x)(1 - P_{\hat{\theta}|X}(p|x)) \\ &= \frac{1}{2} + \frac{1}{2}\sum_x \left(p(x) - q(x)\right) P_{\hat{\theta}|X}(p|x) \\ &\le \frac{1}{2} + \frac{1}{2}\sum_{x:p(x)>q(x)} p(x) - q(x) \\ &= \frac{1}{2} + \frac{1}{2}d_{\mathrm{TV}}(p,q)\end{aligned} \tag{3}$$

So if $d_{\mathrm{TV}}(p,q) < 1 - 2\delta$, then any algorithm cannot succeed in this task with probability $\geq 1 - \delta$. ■

By Fact 2, we know that in order to prove Lemma 1 it suffices to show $d_{\mathrm{TV}}\left(p^{\otimes m}, q^{\otimes m}\right) \leq 1 - 2\delta$ for $m = \mathcal{O}\left(\frac{\log 1/\delta}{d_{\mathrm{H}}^2(p,q)}\right)$. By Fact 3, we have

$$\begin{aligned}d_{\mathrm{TV}}\left(p^{\otimes m}, q^{\otimes m}\right) &\le \sqrt{1 - \left(1 - d_{\mathrm{H}}^2\left(p^{\otimes m}, q^{\otimes m}\right)\right)^2} \\ &= \sqrt{1 - \left(1 - d_{\mathrm{H}}^2\left(p,q\right)\right)^{2m}} \\ &\le 1 - \frac{1}{2}\left(1 - d_{\mathrm{H}}^2\left(p,q\right)\right)^{2m}\end{aligned} \tag{4}$$

Meanwhile, let $m = \frac{1-e^{-1}}{4}\frac{\log 1/\delta}{d_{\mathrm{H}}^2(p,1q)}$, and as we comment in Remark 1, we can safely assume $d_{\mathrm{H}}^2\left(p,q\right) \leq 1 - e^{-1}$, $\delta \leq \frac{1}{16}$. Then by Fact 3, we have

$$\begin{aligned}\left(1 - d_{\mathrm{H}}^2\left(p,q\right)\right)^{2m} &= \left[\left(1 - d_{\mathrm{H}}^2\left(p,q\right)\right)^{\frac{1-e^{-1}}{d_{\mathrm{H}}^2(p,q)}}\right]^{\frac{1}{2}\log 1/\delta} \\ &\ge \exp\left(-\frac{1}{2}\log 1/\delta\right) \\ &= \sqrt{\delta} \\ &\ge 4\delta,\end{aligned} \tag{5}$$

where the second line is due to the inequality $(1+x)^r \geq 1+rx \; \forall x > -1, r \geq 0$ and the monotonicity of power functions. Combining (4) and (5), we obtain $d_{\mathrm{TV}}\left(p^{\otimes m}, q^{\otimes m}\right) \leq 1 - 2\delta$ as desired. ■

2.1.3 Lower Bound for Learning Discrete Distributions

Theorem 2 *For any algorithm that (ϵ, δ)-learns discrete distributions on $[n]$, it requires at least $m = \Omega\left(\frac{n+\log 1/\delta}{\epsilon^2}\right)$ samples.*

Proof of Theorem 2: To prove the lower bound $\Omega\left(\frac{n+\log 1/\delta}{\epsilon^2}\right)$, we can divide it into $\Omega\left(\frac{n}{\epsilon^2}\right)$ and $\Omega\left(\frac{\log 1/\delta}{\epsilon^2}\right)$, and handle them separately, since given these two lower bounds we would have $m = \Omega\left(\max\left(\frac{n}{\epsilon^2}, \frac{\log 1/\delta}{\epsilon^2}\right)\right) = \Omega\left(\frac{n+\log 1/\delta}{\epsilon^2}\right)$.

By the previous discussion, we can derive an $\Omega\left(\frac{\log 1/\delta}{\epsilon^2}\right)$ lower bound with little effort as follows. Consider learning an unknown Bernoulli distribution p from m i.i.d. samples. Let $\mathcal{A}$ be an algorithm that learns p from input $X_1, X_2, \ldots, X_m \sim \text{Bernoulli}(p)$ and output an estimator $\hat{p}$ satisfying

$$\Pr\left(d_{\mathrm{TV}}\left(\hat{p}, p\right) < \epsilon\right) \geq 1 - \delta \tag{6}$$

then we can construct algorithm $\mathcal{B}$ on the same input X^m such that it computes $\hat{p}$ according to $\mathcal{A}$, and accepts if $d_{\mathrm{TV}}\left(\hat{p}, \text{Bernoulli}(\frac{1}{2}+\epsilon)\right) < \epsilon$, rejects otherwise. It is obvious that $\mathcal{B}$ can distinguish Bernoulli$(\frac{1}{2}+\epsilon)$ from Bernoulli$(\frac{1}{2}-\epsilon)$ with probability $\geq 1-\delta$, by invoking Lemma 1, we obtain $m = \Omega\left(\frac{\log 1/\delta}{\epsilon^2}\right)$.

In order to prove the second part of lower bound $\Omega\left(\frac{n}{\epsilon^2}\right)$, we need the following fact.

Fact 4 *The sample complexity of learning a discrete distribution on $[n]$ is at least as many as required for learning a hypercube embedded in Δ^{n-1}.*

This fact is closely related to the well-known Assouad's Lemma [15] and can be seen as a high-dimension analog/generalization of the reduction from distinguishing Bernoulli to learning a Bernoulli. With this fact, we can instead consider the following class of distributions on $[n]$: without loss of generality, we assume n is even (otherwise we can simply set the probability mass in last component to be 0), and partition $[n]$ into $\frac{n}{2}$ buckets, such that the probability mass in each bucket $b_i = \{2i-1, 2i\}, i \in \left[\frac{n}{2}\right]$ is distributed as follows[2]

$$\begin{aligned} p(2i-1) &= \frac{1-100\epsilon z_i}{n} \\ p(2i) &= \frac{1+100\epsilon z_i}{n} \end{aligned} \tag{7}$$

where $z_i \in \{-1, 1\}, \mathbf{z} = \left(z_1, \ldots, z_{\frac{n}{2}}\right)$ is the vertex of Assouad's hypercube. We are going to prove the following lemma

Lemma 2 *Learning a distribution in the above class with probability* $\geq \frac{2}{3}$ *requires* $m = \Omega\left(\frac{n}{\epsilon^2}\right)$ *samples.*

Consider an algorithm $\mathcal{A}$ on input X^m drawn from the class of distributions defined above, computes a vector $\hat{z} \in \{-1, 1\}^{\frac{n}{2}}$, Denote the number of samples falling into bucket b_i as m_i. Clearly the output of $\mathcal{A}$ only depends on the histogram $\{Y_i\}_{i=1}^n$ where $Y_i = \sum_{j=1}^m \mathbb{1}\{X_j = i\}$, since a permutation of X^m will not change the input distribution. Intuitively, conditioned on each bucket b_i, the outcome is a Bernoulli($\frac{1 \pm 100\epsilon}{2}$) random variable. To be more specific, if conditioned on $\{m_i\}_i$, samples outside bucket b_i and $\{z_j\}_{j \neq i}$, all the remaining randomness about event "$\hat{z}_i \neq z_i$" takes place in z_i and algorithm, and from Fact 2 we have $\Pr\left(\hat{z}_i \neq z_i\right) \geq \frac{1}{2} - \mathcal{O}(\epsilon)\sqrt{Y_{b_i}}$.

Then we obtain the inequality $\mathbb{E}\left[\mathbb{1}\{\hat{z}_i \neq z_i\}\right] \geq \frac{1}{2} - \mathcal{O}(\epsilon)\sqrt{Y_{b_i}}$, where the expectation is conditioned on the quantities mentioned before. It immediately follows that

$$\begin{aligned} \mathbb{E}\left[\sum_{i}^{n/2} \mathbb{1}\{\hat{z}_i \neq z_i\}\right] &\geq \mathbb{E}\sum_{i=1}^{n/2} \frac{1}{2} - \mathcal{O}(\epsilon)\sqrt{Y_{b_i}} \\ &= \frac{n}{4} - \mathcal{O}(\epsilon)\sum_i \mathbb{E}\sqrt{Y_{b_i}} \\ &\geq \frac{n}{2} - \mathcal{O}(\epsilon)\sum_i \sqrt{\mathbb{E}\left[Y_{b_i}\right]} \\ &= \frac{n}{2} - \mathcal{O}(\epsilon)\sum_i \sqrt{\frac{2m}{n}} \\ &= n\left(\frac{1}{4} - \mathcal{O}(\epsilon)\sqrt{\frac{m}{2n}}\right) \end{aligned} \tag{8}$$

[2] Here we implicitly require $\epsilon \leq 0.01$. As discussed in Remark 1, such assumption could be made without hurting the asymptotic bounds.

where the third line is due to Jensen's inequality. So for sufficiently small ϵ, we have $\mathbb{E}\left[\sum_{i=1}^{n/2} \mathbb{1}\{\hat{z}_i \neq z_i\}\right] \geq n\left(\frac{1}{4} - c_0\epsilon\sqrt{\frac{m}{2n}}\right)$ with some constant $c_0 > 0$. Now if we set $m = \frac{2n}{(16c_0\epsilon)^2} = \mathcal{O}\left(\frac{n}{\epsilon^2}\right)$, by applying Markov's inequality, we obtain

$$\Pr\left(\sum_{i=1}^{n/2} \mathbb{1}\{\hat{z}_i = z_i\} \geq 0.99\frac{n}{2}\right) \leq \frac{\frac{n}{2} - \mathbb{E}\left[\sum_{i=1}^{n/2} \mathbb{1}\{\hat{z}_i \neq z_i\}\right]}{0.99n/2} \leq \frac{1/2 - 3/16}{0.99/2} < \frac{2}{3}$$

which completes the proof. ■

2.2 Learning Markov Chains: Upper Bounds

We first consider the problem of learning a finite Markov chain from one observed trajectory under the infinity matrix norm. Recalling the definition of infinity matrix norm in (1), to make the estimator $\hat{M}$ close to M, one has to ensure that the outgoing transition probability estimation of every state is close to the ground truth, i.e. $d_{\mathrm{TV}}(\hat{M}_i, M_i)$ is small for all $i \in [n]$. Also note that, due to the memorylessness of Markov chains, the succeeding states of state i follow from an i.i.d. discrete distribution M_i. Therefore, learning the Markov chain requires that the trajectory includes enough number of succeeding states of i to learn M_i. By leveraging this intuition, [1] considers the following notion of k-cover time[3]:

Definition 2 (k-cover time; Definition 4 in [1]) *For any $k \in \mathbb{N}^+$, the random k-cover time $\tau_{cov}^{(k)}$ is the first time when every state in $[n]$ has been visited k times, i.e., $\tau_{cov}^{(k)} \triangleq \inf\{t : \forall i \in [n], N_i(X_1^t) \geq k\}$. And the k-cover time is $t_{cov}^{(k)} \triangleq \max_{i_0 \in [n]} \mathbb{E}[\tau_{cov}^{(k)} | X_0 = i_0]$.*

With the definition of k-cover time at hand, given any (ϵ, δ)-learner $\mathcal{L}(Y_1^m, n)$ for discrete distributions, [1] proposes the following "row-wise" algorithm:

[3] It is worth noting that the k-cover time implicitly depends on the transition matrix M, which is omitted in our notation for simplicity.

Algorithm 1

```
**Input:** Synchronization gap $\tau$, number of nodes $m$, communication rounds $S$, training $v$ steps $k$
**for** $s \leftarrow 1, 2, \ldots, S$ **do**
    all nodes send their local model $f_i^{(s\tau)}$ and $u_i^{(s\tau)}$ to server
    $f^{(s\tau)} = \frac{1}{m}\sum_{i=1}^{m} f_i^{(s\tau)}$
    $u^{(s\tau)} = \frac{1}{m}\sum_{i=1}^{m} u_i^{(s\tau)}$
    server sends $f^{(s\tau)}$ $u^{(s\tau)}$ to all nodes
    all nodes updates $f_i^{(s\tau)} \leftarrow f^{(s\tau)}$ and $u_i^{(s\tau)} \leftarrow u^{(s\tau)}$
    **for** $i \leftarrow 1, \ldots, m$ **do**
        **for** $t \leftarrow s\tau, \ldots, (s+1)\tau - 1$ **do**
            train $v_i$ for $k$ steps using adam (ascent)
            $u_i^{(t+1)} \leftarrow Adam\_ascent(u_i^{(t)})$
            $f_i^{(t+1)} \leftarrow SGD(f_i^{(t)})$
        **end for**
    **end for**
**end for**
```

Data set: MNIST; Network f: 2-layer MLP with 200 neurons and ELU activation; Loss: cross-entropy loss; Optimization task:

$$\min_{f} \max_{u \in \mathbb{R}^{784 \times r}} \frac{1}{n} \sum_{i=1}^{n} \max_{v_i \in \mathbb{R}^r} l(f(x_i + uv_i), y_i)$$

Federated learning setting: 100 nodes, each node has 300 figures with one label and 300 figures with another label
[1] provides theoretical guarantees on the performance of Algorithm 1:

Lemma 3 (Upper bound on learning Markov chains; Lemma 6 in [1]) *If we have a (ϵ, δ)-learner for n-state discrete distribution with sample complexity $k(n, \epsilon, \delta) = \Theta(\frac{n + \log 1/\delta}{\epsilon^2})$,*[4] *then there exists a (ϵ, δ)-learner for the Markov chain M using $\mathcal{O}_\delta(t_{\text{cov}}^{(k(n,\epsilon,\delta/2n))})$ samples. Here $\mathcal{O}_\delta(\cdot)$ hides logarithmic factors in δ.*

Proof Sketch of Lemma 3: When the length of trajectory is $\mathcal{O}_\delta(t_{\text{cov}}^{(k(n,\epsilon,\delta/2n))})$, one could prove that with probability $\geq 1 - \delta/2$, every state $i \in [n]$ is visited

[4] We obtain this bound by combining Theorems 1 and 2.

for $N_i(X^m) > k(n, \epsilon, \delta/2n)$ times, thus i has at least $k(n, \epsilon, \delta/2n)$ succeeding states[5]. With this at hand, one can employ the (ϵ, δ)-learner $\mathcal{L}$ to learn the outgoing transition distribution of every state such that the error under total variation distance is $< \epsilon$ with probability $\geq 1 - \delta/2n$. Invoking the memorylessness of Markov chains, by union bound, this algorithm can learn M with error $< \epsilon$, with probability $\geq 1 - \delta$. ■

Arising from this algorithm, [1] raises the following open problem: whether one can prove corresponding lower bounds on sample complexities such that this upper bound is tight up to a constant factor? In the following part of this work, we give an affirmative answer to this open problem.

2.3 Differential Privacy for Dependent Data

In the context of computational statistics, we may perform learning or testing on sensitive data (e.g. medical records or other behavioral phenomena), and it is desirable to perform the learning or testing tasks without disclosing the sensitive information. One promising way is via differential privacy [16]. In our context, a dataset is a multiset of samples drawn from a distribution over $[n]$, and we say two datasets X^m, Y^m are neighboring datasets if they differ in exactly one element. The (standard) differential privacy is defined as follows:

Definition 3 (Differential privacy) *A randomized algorithm $\mathcal{A} : [n]^m \to \Omega$ is said to be ξ-differentially private if for any $S \subseteq \Omega$ and any neighboring datasets X^m, Y^m, it satisfies* $\Pr[\mathcal{A}(X^m) \in S] \leq \exp(\xi) \Pr[\mathcal{A}(Y^m) \in S]$.

Note that the privacy guarantee of standard differential privacy can deteriorate without independence assumptions (see e.g. [17, 18]). A simple example is to compute mean statistics for each of r rows in a database, namely, $\mu_1, \ldots, \mu_r$, with $\mu_i \in \mathbb{R}$, while enforcing the constraint that any consecutive pairs must satisfy $\mu_i + \mu_{i+1} = c_i$ for some publicly known $c_i \in \mathbb{R}$ $\forall i \in [r-1]$. Then even if independent noises are added to each randomized response for the value μ_j, the attacker can leverage the dependency to get an estimator of every μ_j with arbitrarily small variance, so the privacy guarantee of differential privacy is compromised.

[5] It is worth noting that in a single trajectory, state i has $N_i(X^m)$ succeeding states, except for the last state having $N_i(X^m) - 1$.

To provide stronger privacy guarantee for dependent differential privacy, [11] and [12] proposed equivalent definition of dependent differential privacy[6], which can be stated as follows

Definition 4 (Dependent differential privacy) *A randomized algorithm $\mathcal{A} : [n]^m \to \Omega$ on input $(x_1, x_2, \ldots, x_m)$ is said to be ξ-dependent differentially private if for every $S \subseteq \Omega, x_i, x'_i \in [n], K \subseteq [m] \setminus \{i\}$ (and its complement under $[m]\setminus\{i\}$, denoted as K^C), such that $\mathbb{P}[x_i, x_K], \mathbb{P}[x'_i, x_K] > 0$, it satisfies $\mathbb{P}[\mathcal{A}(x_i, x_K, X_{K^C}) \in S] \leq \exp(\xi)\, \mathbb{P}[\mathcal{A}(x'_i, x_K, X_{K^C}) \in S]$, where*

$$\mathbb{P}[\mathcal{A}(x_i, x_K, X_{K^C}) \in S] = \sum_{x_{K^C}} \mathbb{P}[x_{K^C} | x_i, x_K] \Pr[\mathcal{A}(x_i, x_K, x_{K^C}) \in S]$$

.

Note that dependent differential privacy (DDP) is a strictly stronger definition than standard differential privacy (DP) because DDP degenerates back to DP if we take $K = [m] \setminus \{i\}$.

Definition 5 (**(ϵ, ξ)-DDP Tester**) *Consider an algorithm $\mathcal{A}$ that takes independent samples (respectively a Markovian trajectory) X^m from a discrete distribution (resp. a Markov chain) M over n states with a target distribution $\tilde{M}$ (resp. a Markov chain), and outputs "Accept" or "Reject". $\mathcal{A}$ is said to be a (ϵ, ξ)-DP (resp. (ϵ, ξ)-DDP) tester for discrete distributions (resp. Markov chains) with sample complexity $k(n, \epsilon, \xi)$, if it is ξ-differentially private (resp. ξ-dependent differentially private) and outputs "Accept" if $M = \tilde{M}$; "Reject" if $d_{TV}(M, \tilde{M}) \geq \epsilon$ (resp. $||M - \tilde{M}||_\infty \geq \epsilon$), with probability $\geq \frac{4}{5}$.*

Remark 2 *The notion of (ϵ, ξ)-DDP tester could also be extended to general dependent distributions, with a proper definition of the distance between two distributions. However, as it is out of the scope of this work, we only focus on Markov chains here.*

[6] In [11] it is called "Bayesian differential privacy".

3 Main Results

As described in this Section, we have two main results, both regarding lower bounds on learning and testing dependent data. The first result, Theorem 3 in Section 3.1, solves an open problem proposed in [1] – proving the order-wise optimality of the sample complexity of Algorithm 1 (Algorithm 1 in [1]). The second result, Theorem 4 in Section 3.2, shows an impossibility result – a non-trivial constant lower bound on the dependent differential privacy parameter ξ; and a subsequent result, Corollary 1, gives a similar result with respect to Markov chains.[7]

3.1 Lower Bound on Learning Markov Chains

Recalling that Algorithm 1 is an (ϵ, δ)-learner for Markov chains with sample complexity $\mathcal{O}_\delta(t_{\text{cov}}^{(k(n,\epsilon,\delta/2n))})$. Here we claim that this complexity is order-wise optimal:

Theorem 3 (Lower bound on learning Markov chains) *For any (ϵ, δ)-learner for the Markov chain M, its sample complexity is at least $\Omega(t_{cov}^{(k(n,\epsilon,\delta/2n))})$.*

3.2 Lower Bound on Dependent Differential Privacy

Despite the fact that dependent differential privacy is a natural extension of differential privacy and provides a strong privacy guarantee for statistical estimation on dependent data, our hardness result indicates that, unfortunately, DDP is not suitable for tasks like learning and testing, because those tasks are inherently non-private in DDP sense, and the privacy parameter ξ cannot be arbitrarily small. Therefore, it is also not a suitable privacy definition for learning and testing Markov chains, as a corollary.

Theorem 4 (Lower bound on the worst case DDP) *For any randomized algorithm $\mathcal{A}$ that can distinguish two distinct joint distributions on $[n]^m$ with small error probability, its DDP parameter must satisfy $\xi > \Omega(1)$.*

Corollary 1 (Lower bound on testing Markov chains under DDP) *Any (ϵ, ξ)-DDP tester for a Markov chain with n states must satisfy $\xi > \Omega(1)$.*

[7] As this is a final year project report, it is worth mentioning again that all the proofs in Sections 4 and 5 are our original work, except for Lemma 5.

4 Lower Bound on Learning Markov Chains

In this section, we prove Theorem 3. More precisely, it is equivalent to proving the following: there exist constants $c_1, c_2 > 0$, such that for any learner $\mathcal{L}$ of Markov chain M with access to a single trajectory X^m and the number of states n,

$$\Pr\left(||M - \mathcal{L}(X^m, n)||_\infty \geq \epsilon \mid m \leq c_1 \cdot t_{\text{cov}}^{k(n,\epsilon,\delta/2n)}\right) > c_2, \tag{9}$$

It is worth noting that (9) indicates that $\mathcal{L}$ cannot (ϵ, δ)-learn M for arbitrary small δ, thus proves Theorem 3.

We first provide some intuition regarding the proof. Note that estimating a Markov chain M with small error under the matrix infinity norm requires estimating every outgoing transition distribution with small error, i.e., $||M_i - \hat{M}_i||_1 \leq \epsilon$ for all $i \in [n]$. One straightforward idea is to find a "vulnerable" state, whose transition distribution is difficult to learn. As one can see from Section 2.1, the estimation is more accurate when more samples are observed. Hence, the most "vulnerable" state is intuitively the state with the minimal occurrences of succeeding states. Based on this intuition, we shall prove that when $m = \mathcal{O}(t_{\text{cov}}^{k(n,\epsilon,\delta/2n)})$, it is likely to exist one state with $\mathcal{O}(k(n, \epsilon, \delta/2n))$ occurrences, thus any algorithm fails with constant probability to learn such state's transition distribution (see (11) later).

Denote $N_i \triangleq N_i(X^{m-1})$, which is the number of state i's succeeding states in the trajectory X^m. For simplicity, denote $k \triangleq k(n, \epsilon, \delta/2n)$, $t \triangleq t_{\text{cov}}^{k(n,\epsilon,\delta/2n)}$, and $\hat{M} \triangleq \mathcal{L}(X^m, n)$. Without loss of generality, assume that $N_1 = \min_{i\in[n]} N_i$. As we are interested in asymptotic bounds, we can assume that n is even[8]. Following from (7) in Section 2.1.3, assume that $\lambda \triangleq 100\epsilon$, and consider that the Markov chain is drawn randomly from the following set

$$\mathcal{S} \triangleq \left\{ M : M_{i,2j-1} = \frac{1 - \lambda z_{ij}}{n} \text{ and } M_{i,2j} = \frac{1 + \lambda z_{ij}}{n} \right. \\ \left. \text{with } z_{ij} \in \{-1, 1\},\ \forall i \in [n],\ j \in [n/2] \right\}. \tag{10}$$

[8] Recall similar consideration made in Section 2.1.3.

Considering a constant $c_3 > 0$, we can derive

$$\begin{aligned}
&\Pr\left(||M - \hat{M}||_\infty \geq \epsilon \mid m \leq c_1 t\right) \\
&= \Pr\left(\max_{i\in[n]} ||M_i - \hat{M}_i||_1 \geq \epsilon \mid m \leq c_1 t\right) \\
&\geq \Pr\left(||M_1 - \hat{M}_1||_1 \geq \epsilon \mid m \leq c_1 t\right) \\
&\geq \Pr\left(||M_1 - \hat{M}_1||_1 \geq \epsilon,\ N_1 < c_3 k \mid m \leq c_1 t\right) \\
&= \Pr\left(||M_1 - \hat{M}_1||_1 \geq \epsilon \mid N_1 < c_3 k,\ m \leq c_1 t\right) \cdot \Pr\left(N_1 < c_3 k \mid m \leq c_1 t\right),
\end{aligned} \tag{11}$$

We start with bounding the first part of (11) by the following Lemma:

Lemma 4 (Lower bound on learning one transition distribution)[9] *There exists universal constants $c_1, c_3 > 0$ such that, considering the following feasible set of trajectories*

$$\mathcal{F} \triangleq \left\{X^m : X^m \sim M \in \mathcal{S},\ N_1 = \min_{i\in[n]} N_i < c_3 k,\ m \leq c_1 t\right\}, \tag{12}$$

for any learner $\mathcal{L}$ that learns Markov chain M from its trajectory X^m and outputs $\hat{M}$, if X^m is in the feasible set $\mathcal{F}$, then the learner fails to learn the transition distribution from state 1 with non-vanishing probability, i.e., $\exists\, p_1 > 0$ such that

$$\Pr\left(||M_1 - \hat{M}_1||_1 \geq \epsilon \mid X^m \in \mathcal{F}\right) > p_1. \tag{13}$$

Proof of Lemma 4: We prove this Lemma by reducing the problem to learning discrete distributions from i.i.d. samples. Consider a distribution p such that for all $j \in [n/2]$, $p_{2j-1} = \frac{1-\lambda z_j}{n}$ and $p_{2j} = \frac{1+\lambda z_j}{n}$, where $\lambda = 100\epsilon$ and z_j is randomly drawn from $\{-1, 1\}$. In order to prove the Lemma by using Theorem 2, given infinitely many i.i.d. samples $S_1, S_2, \ldots$ drawn from p, we develop a learner for p by invoking $\mathcal{L}$:

[9] The proof of this lemma is partly guided by the project supervisor, Prof. Siu On Chan.

Algorithm 2 Learning discrete distribution p by invoking $\mathcal{L}$

```
 1: Input: samples $S_1, S_2, \ldots$
 2: Output: a candidate distribution $\hat{p}$
 3: Initialize $M'$ to be an empty $n \times n$ matrix
 4: for $i \in [n]\backslash\{1\}$, $j \in [n/2]$ do
 5:     Randomly draw $z_{ij} \sim \text{Uniform}\{-1, 1\}$
 6:     $M'_{i,2j-1} \leftarrow \frac{1-\lambda z_{ij}}{n}$, $M'_{i,2j} \leftarrow \frac{1+\lambda z_{ij}}{n}$
 7: end for
 8: repeat
 9:     Initialize $X^m$ to be empty, $X_1 \leftarrow 2$
10:     for $num \in [m-1]$ do
11:         if $X_{num} = 1$ then
12:             $X_{num+1} \leftarrow S_{N_1(X^{num})}$
13:         else
14:             Randomly draw $X_{num+1} \leftarrow \text{Multinomial}(M'_{X_{num}})$
15:         end if
16:     end for
17:     if $N_1(X^{m-1}) \neq \min_{i\in[n]} N_i(X^{m-1})$ or $N_1(X^{m-1}) \geq c_3 k$ then
18:         $X^m \leftarrow None$
19:     end if
20: until $X^m \neq None$
21: Invoke $\hat{M} \leftarrow \mathcal{L}(X^m, n)$
22: return $\hat{p} \triangleq \hat{M}_1$
```

First of all, it is trivial that $\mathcal{F}$ is feasible and Algorithm 2 ends with probability 1, by considering the case of $N_1 = 0$. Denote Markov chain $M \triangleq (p, M'_2, M'_3, \ldots, M'_n)$. One could easily see that Algorithm 2 without steps 17-19 generates a Markovian trajectory distributed as M of length m; while steps 17-19 discard all non-feasible trajectories such that $N_1(X^{m-1}) \neq \min_{i\in[n]} N_i(X^{m-1})$ or $N_1(X^{m-1}) \geq c_3 k$. Combining these two facts, we obtain that the trajectory drawn from steps 8-20 is feasible, i.e., $X^m \in \mathcal{F}$.

On the other hand, note that conditioned on the event $X^m \in \mathcal{F}$, Algorithm 2 is a learner for the discrete distribution p with access to $< c_3 k$ samples drawn from p (since any trajectory such that $N_1(X^{m-1}) \geq c_3 k$ is discarded). From Theorem 2 we know that there exist constants $c_4, p_1 > 0$, such that when sample complexity $< c_4 k(n, \epsilon, \delta)$[10], any learning algorithm

[10] Recalling that $k(n, \epsilon, \delta) = \Theta(\frac{n+\log 1/\delta}{\epsilon^2})$ as explained in Lemma 3.

fails with probability $> p_1$. Note that

$$k = k(n, \epsilon, \delta/2n) = \Theta\left(\frac{n + \log(2n/\delta)}{\epsilon^2}\right) = \Theta\left(\frac{n + \log 1/\delta}{\epsilon^2}\right) = \Theta(k(n, \epsilon, \delta)).$$

Hence there exists constant $c_3 > 0$ such that $c_3 k \leq c_4 k(n, \epsilon, \delta)$. By Theorem 2 we obtain that

$$\Pr\left(||M_1 - \hat{M}_1||_1 \geq \epsilon \mid X^m \in \mathcal{F}\right) > p_1, \tag{14}$$

thus finishing the proof of Lemma 4. ■

We then bound the second part of (11). For simplicity, denote $\tau \triangleq \tau_{\text{cov}}^{(k)}$ and $\tau_1 \triangleq \tau_{\text{cov}}^{(c_3 k)}$ (recalling Definition 2). First note that

$$\begin{aligned}\Pr\left(N_1 < c_3 k \mid m \leq c_1 t\right) &= 1 - \Pr\left(N_1 \geq c_3 k \mid m \leq c_1 t\right) \\ &= 1 - \Pr\left(\min_{i \in [n]} N_i \geq c_3 k \mid m \leq c_1 t\right) \\ &= 1 - \Pr\left(\tau_1 \leq c_1 t\right),\end{aligned} \tag{15}$$

where the second equality is due to the definition $N_1 = \min_{i \in [n]} N_i$. As we are interested in asymptotic bounds, assume that $\frac{1}{c_3} \in \mathbb{N}$ (otherwise we can substitute it with $\frac{1}{\lceil 1/c_3 \rceil}$). Next, consider dividing a Markovian trajectory of length $\frac{c_1}{c_3} t$ into $1/c_3$ consecutive pieces, each of length $c_1 t$. Due to the memorylessness of Markov chains, given the probability distribution of the initial states of all pieces, the pieces are independent of each other. It is worth noting that, if it holds that every consecutive piece has a $(c_3 k)$-cover subtrajectory, then the whole trajectory must have a k-cover subtrajectory. Hence, by denoting $q_0 \triangleq \arg\min_{q \in \Delta^{n-1}} \Pr(\tau_1 \leq c_1 t \mid \text{initial state} \sim q)$, we obtain that

$$\Pr\left(\tau \leq \frac{c_1}{c_3} t\right) \geq \Pr(\tau_1 \leq c_1 t \mid q_0)^{1/c_3}. \tag{16}$$

Upon assigning $c_1 = c_3/3$, the problem reduces to give an upper bound on $\Pr\left(\tau \leq t/3\right)$. Recalling that $t = \max_{i_0 \in [n]} \mathbb{E}[\tau | X_0 = i_0]$, intuitively, the probability distribution of τ should concentrate around t. We shall introduce the following lemma from [1], which confirms the intuition from the other side and is also useful for subsequent proof:

Lemma 5 (Exponential decay lemma; Lemma 5 in [1] *For random walk on irreducible chains, for any $k, \gamma \in \mathbb{N}^+$, and any initial distribution q, we have* $\Pr\left(\tau_{cov}^{(k)} \geq e\gamma t_{cov}^{(k)}\right) \leq e^{-\gamma}$.

Proof Sketch of Lemma 5: Consider τ with any fixed starting state $X_0 \sim q$, by Markov's inequality, we have that

$$\Pr(\tau \geq et) \leq \Pr(\tau \geq e \cdot \mathbb{E}[\tau | X_0]) \leq 1/e.$$

Note that this inequality holds for any initial distribution q. Then we divide a trajectory of length $e\gamma t$ into γ consecutive pieces of length et. Again, due to the memorylessness of Markov chains, given the probability distribution of the initial states of all pieces, the pieces are independent of each other. Note that, if it holds that the whole trajectory does not have k-cover a subtrajectory, then no piece has a k-cover subtrajectory. Hence $\Pr(\tau \geq e\gamma t) \leq \Pr(\tau \geq et)^{\gamma} \leq e^{-\gamma}$. ■

Lemma 5 shows that with high probability, τ should not be too much larger than t. Upon invoking Lemma 5 with delicate slicing of intervals, we could provide an upper bound on $\Pr(\tau \leq t/3)$:

Lemma 6 (Concentration bound on k-cover time) *For random walk on irreducible chains, for any $k \in \mathbb{N}^+$, and any initial distribution q, we have $\Pr\left(\tau_{cov}^{(k)} \leq t_{cov}^{(k)}/3\right) \leq 0.951$.*

Proof of Lemma 6: For the sake of contradiction, suppose $\Pr(\tau \leq t/3) > 0.951$. Considering slicing $\mathbb{N}^+$ into $[1, t/3]$, $(t/3, 3et)$, and $[e\gamma t, e(\gamma+1)t)$ for all integers $\gamma \geq 3$, we shall have the following:

$$\begin{aligned}
t = \mathbb{E}[\tau] &= \sum_{j=1}^{\infty} \Pr(\tau = j) \cdot j \\
&< \Pr(\tau \in [1, t/3]) \cdot t/3 + \Pr(\tau \in (t/3, 3et)) \cdot 3et \\
&\quad + \sum_{\gamma=3}^{\infty} \Pr(\tau \in [e\gamma t, e(\gamma+1)t)) \cdot e(\gamma+1)t \\
&= \Pr(\tau \leq t/3) \cdot t/3 + (1 - \Pr(\tau \leq t/3) - \Pr(\tau \geq 3et)) \cdot 3et \\
&\quad + \sum_{\gamma=3}^{\infty} (\Pr(\tau \geq e\gamma t) - \Pr(\tau \geq e(\gamma+1)t)) \cdot e(\gamma+1)t \qquad (17) \\
&< 0.951 \cdot t/3 + (1 - 0.951 - e^{-3}) \cdot 3et + \sum_{\gamma=3}^{\infty} (e^{-\gamma} - e^{-\gamma-1}) \cdot e(\gamma+1)t \qquad (18) \\
&= 0.9307t < t,
\end{aligned}$$

where (18) comes from observing that (17) increases as $\Pr(\tau \leq t/3)$ decreases or $\Pr(\tau \geq e\gamma t)$ increases, and using the assumption $\Pr(\tau \leq t/3) > 0.951$ and the upper bound $\Pr(\tau \geq e\gamma t) \leq e^{-\gamma}$ from Lemma 5. Hence we lead to a contradiction, thus it completes the proof. ■

With (15), (16) and Lemma 6 at hand, we are able to upper bound the second part of (11) as follows (recalling that c_1 is defined as $c_3/3$):

$$\begin{aligned}\Pr\left(N_1 < c_3 k \mid m \leq c_1 t\right) &= 1 - \Pr\left(\tau_1 \leq c_1 t\right) \\ &\geq 1 - \Pr\left(\tau \leq \frac{c_1}{c_3} t\right)^{c_3} \\ &\geq 1 - 0.951^{c_3}, \qquad (19)\end{aligned}$$

which is a positive constant because $c_3 > 0$.

Finally, Lemma 4 and Equation (19) give non-vanishing bounds on the two terms of (11) respectively, then we can derive (9), hence finishing the proof of Theorem 3.

5 Lower Bound on Dependent Differential Privacy

5.1 Lower Bound on the Worst Case DDP

Proof of Theorem 4: In this subsection, we prove Theorem 4. For convenience, we change the support set from $\{1, 2, \ldots, n\}$ to $\{0, 1, \ldots, n-1\}$. Consider m samples X^m =$(x_0, x_1, \ldots, x_{m-1})$ drawn from joint distribution Q, given any randomized algorithm $\mathcal{A}$ on input X^m, denote $x_{[m]} = (x_1, x_2, \ldots, x_m)$, and $P_A\left(x_0, x_{[m-1]}\right)$ be the probability of accepting $(x_0, x_{[m-1]})$. Furthermore, we define for every $x_{[m-1]}$,

$$\begin{aligned} P_{A_0}\left(x_{[m-1]}\right) &= P_A\left(0, x_{[m-1]}\right) \\ P_{A_1}\left(x_{[m-1]}\right) &= P_A\left(1, x_{[m-1]}\right) \end{aligned} \tag{20}$$

Note that P_{A_0} and P_{A_1} fully describe the behavior of algorithm $\mathcal{A}$ and are not PMFs, since they only take care of the accepting probabilities.

Now consider the task of distinguishing between distributions Q_0 and Q_1, where $Q_0 = \text{Bernoulli}(\frac{1}{2}) \otimes D_0, Q_1 = \text{Bernoulli}(\frac{1}{2}) \otimes D_1$, and $d_{\text{TV}}(D_0, D_1) \geq d > 0$ for some constant d. Then the output "Accept" means to output "Q_0" instead of "Q_1". Given any randomized algorithm $\mathcal{A}$ that can succeed in this task with high probability, say, $\delta < \frac{1}{5}$, it automatically satisfies

$$\begin{aligned} \mathbb{E}_{X^m \sim Q_0}\left[P_A\left(X_0, X_{[m-1]}\right)\right] &\geq \frac{4}{5} \\ \mathbb{E}_{X^m \sim Q_1}\left[P_A\left(X_0, X_{[m-1]}\right)\right] &\leq \frac{1}{5} \end{aligned} \tag{21}$$

for sufficiently large m.

By construction, we can decompose Q_0 and Q_1 in inequalities (21) into

$$\begin{aligned} \mathbb{E}_{X_{[m-1]} \sim D_0}\left[\frac{1}{2}P_{A_0}\left(X_{[m-1]}\right) + \frac{1}{2}P_{A_1}\left(X_{[m-1]}\right)\right] &\geq \frac{4}{5} \\ \mathbb{E}_{X_{[m-1]} \sim D_1}\left[\frac{1}{2}P_{A_0}\left(X_{[m-1]}\right) + \frac{1}{2}P_{A_1}\left(X_{[m-1]}\right)\right] &\leq \frac{1}{5} \end{aligned} \tag{22}$$

By the boundedness of P_A, it is clear that

$$\begin{aligned} \mathbb{E}_{D_0}\left[P_{A_0}\left(X_{[m-1]}\right)\right] &\geq \frac{3}{5} \\ \mathbb{E}_{D_1}\left[P_{A_1}\left(X_{[m-1]}\right)\right] &\leq \frac{2}{5} \end{aligned} \tag{23}$$

Now consider another distribution $Q_3 = \text{Bernoulli}(p) P_{X_{[m-1]}|X_0}$, where $P_{X_{[m-1]}|X_0}$ satisfies $P_{X_{[m-1]}|X_0=0} = D_0$ and $P_{X_{[m-1]}|X_0=1} = D_1$, then according to the definition 4 (DDP), we have

$$\mathbb{E}_{D_0}\left[P_{A_0}(X_{[m-1]})\right] \leq \exp(\xi)\mathbb{E}_{D_1}\left[P_{A_1}(X_{[m-1]})\right] \tag{24}$$

Comparing equations (23) and (24), we conclude that $\xi \geq \log \frac{3}{2} > 0$. ■

5.2 Lower Bound on Testing Markov Chains under DDP

Proof of Corollary 1: To derive Corollary 1, we start with two Markov chains M_1, M_2 with $||M_1 - M_2||_\infty \geq \Omega(1)$ and trajectory $X^m = (x_1, x_2, \dots, x_m)$. Note that an initial distribution π together with a Markov chain fully specifies a joint distribution on the Markovian trajectory X^m. Denote the joint distribution given by (π, M_1) (resp. (π, M_2)) as $P_1(x_1, x_2, \dots, x_m)$ (resp. as $P_2(x_1, x_2, \dots, x_m)$). We can take two trajectories from these two joint distributions with positive probability, such that they differ in at least one position $\tau \in [m]$. (Such pairs of trajectory always exist, otherwise two joint distributions will be identical, which contradicts our assumption.) Without loss of generality, assume that they differ in the τ-th position and the labels are 1 and 2, respectively. The respective marginal distributions on $[m] \setminus \{\tau\}$ (a trajectory without the sample at time $\tau \in [m]$), are denoted as $D_1 = \sum_{x_\tau} P_1(x_1, \dots, x_m)$, and $D_2 = \sum_{x_\tau} P_2(x_1, \dots, x_m)$.

Then we can construct another joint distribution $Q^m(M_1, M_2, \tau)$ factorized as $Q_{X_\tau} Q_{X_1,\dots,X_m|X_\tau}$[11], where $Q_{X_\tau} = \text{Uniform}\{1, 2\}$ and $Q_{X_1,\dots,X_m|X_\tau}$ satisfies

$$\begin{aligned} Q_{X_1,\dots,X_m|X_\tau=1} &= D_1 \\ Q_{X_1,\dots,X_m|X_\tau=2} &= D_2 \end{aligned} \tag{25}$$

For any randomized algorithm $\mathcal{A}$ that can test Markov chain M_1 with small error probability (say $\delta \leq \frac{1}{5}$), it will accept a trajectory from M_1 and reject a trajectory from M_2 with high probability for sufficiently large m. Therefore, if we fix some $\tau \in [m]$, when the underlying distribution is $Q^m(M_1, M_2, \tau)$

[11] Note that Q^m is not necessarily a joint distribution according to any (time-homogeneous) irreducible Markov chain, but it can still satisfy the requirement of DDP because DDP does not assume the underlying joint distribution is generated by a Markov chain or not.

(i.e. run algorithm $\mathcal{A}$ on a trajectory of samples drawn from the distribution $Q^m(M_1, M_2, \tau)$), by the same argument as in the previous section, we can obtain $\xi > \Omega(1)$. ■

5.3 Remarks on Privately Learning and Testing on Dependent Data

In this subsection, we will discuss the implications of Theorem 4 and Corollary 1 to avoid misunderstanding.

Differential privacy was initially proposed to defend against "differential attacks", where the attacker infers the individual's sensitive data from the outputs of the algorithm on two (or more) datasets that differs in exactly one element. Therefore, the data are not assumed to be randomly generated according to any distributions. However, as discussed in [18], it does not mean that its privacy guarantee holds without any assumption. Instead, its privacy guarantee is only supposed to work for "differential attack", a particular threat model, if you like.

On the other hand, dependent differential privacy (or Bayesian differential privacy) uses a different threat model, where the attacker is assumed to have partial information about the data (i.e. either deterministic or probabilistic dependence among data). For such variant of differential privacy, the "randomness" or joint distribution can be informally viewed as s smaller search space than the exponential-size space under independent assumption. Clearly, such attack is more difficult to protect against than the standard differential attack.

In fact, as mentioned in [11], no protection is known if the attacker uses a different data model from the one specified in the definition of DDP, and our hardness result (Theorem 4) formally proved that *no* asymptotic privacy guarantee in DDP sense is achievable for any useful learning/testing algorithms if the attacker uses some special data model to perform the differential attack. And Corollary 1 basically proved that a good tester of Markov chains is also vulnerable to such attackers.

6 Conclusion

In this joint work, we discussed elementary bounds for learning and testing, and proved our two main contributions, namely, a lower bound on learning irreducible Markov chains and a hardness result on privately testing generic distributions (including Markov chains) under dependent differential privacy. The first result confirms the order-wise optimality of previous work [1] and the second result reveals that the dependent differential privacy is not suitable for learning and testing tasks.

We believe that many of our hardness results can be extended to a broader class of learning and testing problems for irreducible Markov chains, including standard and tolerant closeness testing, tolerant-uniformity testing, tolerant-identity testing, agnostic testing, robust learning and testing, and more. In addition, some potential follow-up works on upper bound could be (privately and non-privately) learning and testing problems on graphical models, including Markov random fields, Gauss-Markov models, Gaussian-Bayesian models, and so on.

References

[1] S. O. Chan, Q. Ding, and S. H. Li, "Learning and testing irreducible markov chains via the k-cover time," in Algorithmic Learning Theory. PMLR, 2021, pp. 458–480.

[2] M. Anthony, P. L. Bartlett, P. L. Bartlett et al., Neural network learning: Theoretical foundations. cambridge university press Cambridge, 1999, vol. 9.

[3] L. Paninski, "A coincidence-based test for uniformity given very sparsely sampled discrete data," IEEE Transactions on Information Theory, vol. 54, no. 10, pp. 4750–4755, 2008.

[4] S.-O. Chan, I. Diakonikolas, P. Valiant, and G. Valiant, "Optimal algorithms for testing closeness of discrete distributions," in Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms. SIAM, 2014, pp. 1193–1203.

[5] I. Diakonikolas, T. Gouleakis, J. Peebles, and E. Price, "Optimal identity testing with high probability," arXiv preprint arXiv:1708.02728, 2017.

[6] J. Acharya, Z. Sun, and H. Zhang, "Differentially private testing of identity and closeness of discrete distributions," Advances in Neural Information Processing Systems, vol. 31, 2018.

[7] C. L. Canonne, "A survey on distribution testing: Your data is big. but is it blue?" Theory of Computing, pp. 1–100, 2020.

[8] G. Wolfer and A. Kontorovich, "Minimax learning of ergodic markov chains," in Proceedings of the 30th International Conference on Algorithmic Learning Theory, ser. Proceedings of Machine Learning Research, A. Garivier and S. Kale, Eds., vol. 98. PMLR, 22–24 Mar 2019, pp. 904–930.

[9] Y. Hao, A. Orlitsky, and V. Pichapati, "On learning markov chains," in Advances in Neural Information Processing Systems, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018.

[10] Y. Han, S. Jana, and Y. Wu, "Optimal prediction of markov chains with and without spectral gap," Advances in Neural Information Processing Systems, vol. 34, pp. 11 233–11 246, 2021.

[11] B. Yang, I. Sato, and H. Nakagawa, "Bayesian differential privacy on correlated data," in Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, ser. SIGMOD '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 747–762. [Online]. Available: https://doi.org/10.1145/2723372.2747643

[12] J. Zhao, J. Zhang, and H. V. Poor, "Dependent differential privacy for correlated data," in 2017 IEEE Globecom Workshops (GC Wkshps), 2017, pp. 1–7.

[13] J. Lee, "Lecture notes in sublinear algorithms for big data," https://cs.brown.edu/courses/csci1951-w/lec/lec%2012%20notes.pdf, October 2020.

[14] M. Mitzenmacher and E. Upfal, Probability and Computing: Randomized Algorithms and Probabilistic Analysis. USA: Cambridge University Press, 2005.

[15] B. Yu, Assouad, Fano, and Le Cam. New York, NY: Springer New York, 1997, pp. 423–435. [Online]. Available: https://doi.org/10.1007/978-1-4612-1880-7_29

[16] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in Theory of Cryptography, S. Halevi and T. Rabin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 265–284.

[17] C. Liu, S. Chakraborty, and P. Mittal, "Dependence makes you vulnberable: Differential privacy under dependent tuples." in NDSS, vol. 16, 2016, pp. 21–24.

[18] D. Kifer and A. Machanavajjhala, "No free lunch in data privacy," in Proceedings of the 2011 ACM SIGMOD International Conference on Management of data, 2011, pp. 193–204.