Private and Communication-Efficient Algorithms for Entropy Estimation

Gecia Bravo-Hermsdorff
Department of Statistics
University College London
gecia.bravo@gmail.com

Róbert Busa-Fekete
Google Research
busarobi@google.com

Mohammad Ghavamzadeh
Google Research
ghavamza@google.com

Andres Munoz Medina
Google Research
ammedina@google.com

Umar Syed
Google Research
usyed@google.com

Abstract

Modern statistical estimation is often performed in a distributed setting where each sample belongs to a single user who shares their data with a central server. Users are typically concerned with preserving the privacy of their samples, and also with minimizing the amount of data they must transmit to the server. We give improved private and communication-efficient algorithms for estimating several popular measures of the entropy of a distribution. All of our algorithms have constant communication cost and satisfy local differential privacy. For a joint distribution over several variables whose conditional independence given by a tree, we describe algorithms for estimating the Shannon entropy that require a number of samples that is linear in the number of variables, compared to the quadratic sample complexity of prior work. We also describe an algorithm for estimating the Gini entropy whose sample complexity has no dependence on the support size of the distribution and can be implemented using a single round of concurrent communication between the users and the server. In contrast, the previously best-known algorithm has high communication cost and requires the server to facilitate interaction between the users. Finally, we describe an algorithm for estimating the collision entropy that matches the space and sample complexity of the best known algorithm but generalizes it to the private and communication-efficient setting.

1 Introduction

Statistical estimation has traditionally focused on minimizing the number of samples needed to estimate properties of a distribution. In the 'big data' era, statisticians and computer scientists have also tried to minimize the space complexity of estimation algorithms, particularly in the streaming setting. More recently, the increasing prevalence of mobile computing has led to a focus on the privacy and communication costs of statistical estimation. In this paper, we focus on the following setting: a set of users each draw one sample from a distribution each, and share information about their sample with a central server. The central server then uses the collected data to estimate a property of the distribution. Users are concerned with preserving the privacy of their sample, and also with minimizing the amount of data that is transmitted to the server.

For example, consider the problem of detecting fingerprinting on the web [30]. Many websites track users across the web without their consent by recording (enough) information about their devices (e.g., installed fonts, operating system, timezone, etc.), a practice known as "browser fingerprinting". Entropy is the standard metric used to quantify the identifiability of the collected fingerprints. So a

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

private and distributed method for estimating entropy can be used by a browser to warn users that this covert tracking could occur, without ever storing the fingerprints themselves.

The study of entropies has an extensive and rich history in mathematics and sciences. Related quantities called "entropy" appear in many contexts (thermodynamics, information theory, dynamic systems [34], category theory [10, 6], etc.). These may be broadly thought as measures of information of a system or process obeying certain properties, which, in turn, lead to natural measures of disorder, randomness, outcome diversity, information content, uniformity, etc.

In this paper, we study private and communication-efficient algorithms for estimating certain entropies of a distribution. Specifically, we give algorithms for estimating the following entropies, which are widely-used in many scientific fields to quantify the uncertainty, diversity and spread of a discrete distribution:

- ***Shannon entropy*** [32], a fundamental quantity in information theory.
- ***Gini entropy*** (also known as Tsallis entropy [37] of order 2, or (one minus the) second frequency moment). Some of its applications include measuring ecological diversity [33, 27], market competition among firms [22], effective size of political parties [26], and suitability of features to split on during decision tree learning [29].
- ***Collision entropy*** (also known as Rényi entropy [31] of order 2). Some of its applications include measuring the quality of random number generators [35], and determining the number of reads needed to reconstruct a DNA sequence [28].

Our algorithms are implemented in either the *non-interactive* model (for the Gini and collision entropies), in which all users simultaneously exchange information with the server during a single round of communication, or the (stronger) *sequentially interactive* model (for the Shannon entropy), in which the server queries users one at a time, possibly in an adaptive manner [23]. When analyzing the communication complexity of an algorithm, we prove bounds on the number of bits that each user transmits to the server. However, the server is allowed to broadcast an arbitrary amount of information to the users (this is also called the *blackboard* model [17]), including shared random bits (also known as the *public coin* model [1, 2, 24]). When analyzing the privacy of our algorithms, we use the framework of *local differential privacy* [18] which ensures that the server learns very little about each user's data.

Our contributions:

- A sequentially interactive α-local differentially private algorithm for estimating the Shannon entropy of a joint distribution on d variables within ϵd error using $\tilde{O}(d/\alpha^2\epsilon^3)$ samples and $O(1)$ bits per sample. Our analysis assumes that each of the d variables has a constant support size and that their conditional independence graph is a tree. We also describe algorithms that have better dependence on $1/\epsilon$ in certain special cases, such as when the tree has low diameter or is a chain. Our algorithms achieve $O(1)$ communication complexity by observing only two or three of the d variables in any single sample; we call these *pair* and *triplet observations*. The only previously known algorithm for estimating the Shannon entropy of a tree-structured distribution from *pair* observations is a non-interactive algorithm due to Chow and Liu [13]. We also prove that any non-interactive algorithm requires $\Omega(d^2)$ *pair* observations to achieve $O(d)$ error. We also prove that, for any sequentially interactive algorithm, $\Omega(d/\epsilon)$ pair observations are necessary to achieve $O(\epsilon d)$ error.
- A non-interactive α-local differentially private algorithm for estimating the Gini entropy of a distribution within ϵ error using $O(b^2 \max\{1-g, 2^{-b}\}/(\alpha^2\epsilon^2))$ samples, b bits per sample, and $\tilde{O}(b)$ space, where $g \in [0,1]$ is the Gini entropy of the distribution. The best previous algorithm [11] has the same sample complexity, even when $b = 1$, but is sequentially interactive, and also requires $\Omega(k)$ bits per sample and $\Omega(k)$ space, where k is the support size of the distribution. Also our error bound also holds with high probability instead of only in expectation.
- A non-interactive α-local differentially private algorithm for estimating the collision entropy of a distribution with support size k within ϵ error using $\tilde{O}(b^2k^2/(\alpha^2\epsilon^2 \min\{k, 2^b\}))$ samples, b bits per sample, and $\tilde{O}(b)$ space. Setting $b = \log k$ and $\alpha = O(1)$ recovers the sample and space complexity guarantees of the non-interactive algorithm from [35] up to logarithmic factors, and thus, our algorithm generalizes the previously best known algorithm to the private and communication-efficient setting.

2 Related Work

There is a very extensive literature on distributed statistical estimation under communication constraints (see [39] for the paper that appears to have started this thread). Variations on the problem include whether communication is allowed between users, whether communication happens in one or multiple rounds, whether there is a shared source of randomness among the users, and whether communication is limited per-user or only cumulatively across all users.

Many previous results in this area bound the sample and communication complexity of estimating the parameters of a distribution P_θ, where $\theta \in \Theta$ (see *e.g.* [21]). This problem class includes discrete distribution estimation, where the guarantees are usually stated as bounds on the relative entropy or total variation distance between the estimated and true distribution (see *e.g.* [5]). Other problems of interest are mean estimation [36] and heavy hitter estimation [4].

There has also been significant interest in differentially private statistical estimation, and of particular relevance is work by [3], who gave private algorithms for estimating certain functionals of a distribution, including the Shannon entropy. However, they used the central model of differential privacy, while in this paper we prove guarantees using the (stronger) local model.

3 Entropy Measures

The *Shannon*, *Tsallis*, and *Rényi* entropy of a discrete random variable X are defined as

$$\text{(Shannon)} \qquad H(X) = -\sum_x \Pr[X = x] \log \Pr[X = x], \tag{1}$$

$$\text{(Tsallis)} \qquad T_\gamma(X) = \frac{1}{\gamma - 1}\big(1 - \sum_x \Pr[X = x]^\gamma\big), \tag{2}$$

$$\text{(Rényi)} \qquad R_\gamma(X) = \frac{1}{1 - \gamma} \log\big(\sum_x (\Pr[X = x])^\gamma\big), \tag{3}$$

where γ in (2) and (3) is a free parameter satisfying $\gamma > 0$ and $\gamma \neq 1$. Both Tsallis and Rényi entropy are generalizations of Shannon entropy in the sense that $\lim_{\gamma \to 1} T_\gamma(X) = \lim_{\gamma \to 1} R_\gamma(X) = H(X)$.

In this paper, we describe algorithms for estimating the Shannon entropy and special cases of the Tsallis and the Rényi entropy that are widely used in many scientific fields: $T_2(X)$, also called the *Gini entropy*, and $R_2(X)$, also called the *collision entropy*. Substituting $\gamma = 2$ into the definitions above and using the abbreviations $G(X) \equiv T_2(X)$ and $C(X) \equiv R_2(X)$ we have

$$\text{(Gini)} \qquad G(X) \equiv T_2(X) = 1 - \sum_x \Pr[X = x]^2,$$

$$\text{(Collision)} \qquad C(X) \equiv R_2(X) = -\log\big(\sum_x \Pr[X = x]^2\big).$$

Gini entropy is so-called because it is equivalent to Gini diversity index [20], a statistics proposed by Corrado Gini in 1912 [20] to measure income and wealth inequality. Collision entropy takes its name from the observation that if X and X' are independent and identically distributed then $C(X) = -\log \Pr[X = X']$.

For the problem of estimating the Shannon entropy, we specialize to a high-dimensional setting, where we only observe a *pair* (or *triplet*) of the dimensions at a time. That is, X is a random-vector of d discrete variables, where d is large, but each X_i has a constant support size (e.g., the are binary), and we only observe two (or three) dimensions per sample. Without making any assumption about this joint distribution, the problem is intractable. One of the most common assumptions, which we also adopt in this work, is that the joint distribution is tree-structured. In this case, the distribution can be estimated by the celebrated [13] (and optimal [8]) Chow-Liu algorithm. While the Chow-Liu algorithm requires $\Omega(d^2)$ *pairs* observations to estimate the Shannon entropy, our sequential algorithm requires only $\mathcal{O}(d)$ *pairs* observations (see Section 5.2 for more details).

The *joint Shannon entropy* $H(X_1, \ldots, X_d)$ of a set of random variables $X_1, \ldots X_d$ is the Shannon entropy $H(X)$ of the random variable $X = (X_1, \ldots, X_d)$. We write the abbreviated term *joint entropy* when the use of Shannon entropy is obvious from context.

The *mutual information* between two random variables X and Y and their *conditional mutual information* given another random variable Z, which are defined as

$$I(X;Y) = H(X) + H(Y) - H(X,Y), \tag{4}$$

$$I(X;Y \mid Z) = H(X,Z) + H(Y,Z) - H(X,Y,Z) - H(Z). \tag{5}$$

4 Estimation Algorithms and Evaluation Criteria

A set of n users and a central server cooperate according to the following protocol to estimate the entropy of a random variable X:

1. Each user $i \in [n]$ draws an independent sample x_i according to the distribution of X.
2. For r rounds:
   (a) The server sends information to a subset of the users.
   (b) Those users send (partial) information about their sample back to the server.
3. The server outputs an estimate of the Shannon entropy (Algorithm 1,2 and 4) or the Gini or collision (Algorithm 5) entropies of X.

An *estimation algorithm* specifies the steps that each user and the server perform to implement the above protocol. The algorithm is *non-interactive* if the protocol consists of a single round in which all users participate. In a non-interactive algorithm the server cannot adapt its queries to users based on responses from other users, since the server communicates with all the users concurrently. An algorithm is *sequentially interactive* if each round consists of communication with a single user, who is never contacted again. Sequential interactivity enables the server to query users adaptively [23].

We evaluate estimation algorithms according to the following criteria:

- *Sample complexity*: The number of users from whom the server requests data.
- *Space complexity*: The space used by the server when executing the algorithm.
- *Communication complexity*: The maximum number of bits transmitted by any single user to the server. Note that the amount of information sent by the server to the users is not counted when determining communication complexity.
- *Privacy*: Let x_i be the sample belonging to user i and o_i be the data observed by the server from user i. We say that an algorithm satisfies *α-local differential privacy* if

  $$\Pr[o_i \in O \mid x_i = x] \leq e^{\alpha} \Pr[o_i \in O \mid x_i = x']$$

  for any user i, measurable set O, and possible sample values x, x'.
- *Error*: The absolute difference between the true entropy of the distribution and the estimate output by the server.

5 Estimating the Shannon Entropy of Tree-structured Distributions

In this section we assume that $X = (X_1, \ldots, X_d)$ is a vector of d discrete variables, and that the support size of each variable X_i is a constant (*e.g.*, each variable is Boolean). We also assume that X has a *tree-structured* distribution, which means that there exists a rooted tree T with d nodes such that for any node $i \in [d]$ we have $\Pr[X_i \mid X_{-i}] = \Pr[X_i \mid X_{\mathrm{pa}_T(i)}]$, where $X_{-i} = (X_1, \ldots X_{i-1}, X_{i+1}, \ldots, X_d)$ and the node $\mathrm{pa}_T(i)$ is the parent of node i in tree T. If i is the root node, then we define $\Pr[X_i \mid X_{\mathrm{pa}_T(i)}] = \Pr[X_i]$. Equivalently, a tree-structured distribution is a Markov random field with a tree as the underlying undirected graph. Essentially, the tree-structured assumption implies that the only correlations among the X_i's are pairwise correlations. If T is a chain or a star we say that X is *chain-structured* and *star-structured*, respectively. We will treat these two special cases at the end of this section (Algorithms 2 and 4, respectively).

5.1 Breaking-up the problem: Entropy Estimation When the Support Size is Small

Before proceeding to describe algorithms for estimating the Shannon entropy of tree-structured distributions, we use existing results for private distribution estimation to devise a local differentially private estimator for the Shannon entropy that is sample and communication efficient when the support size of the distribution is small (as is the case for the marginals). The server will repeatedly invoke this algorithm as a subroutine in the sections below.

First we recall that the Shannon entropy can be upper bounded according to Theorem 17.3.3 of [14] as

$$|H(X_{\mathbf{p}}) - H(X_{\mathbf{p}'})| \leq \|\mathbf{p} - \mathbf{p}'\|_1 \log \frac{c}{\|\mathbf{p} - \mathbf{p}'\|_1}$$

where $X_{\mathbf{p}}$ and $X_{\mathbf{p}'}$ are two discrete random variables with support size c and distributions $\mathbf{p}$ and $\mathbf{p}'$. Next, we apply a local differentially private learning algorithm for discrete distribution due to [4] that learns the parameters of a discrete distribution with small L_1 error. The following theorem combines these two results by using the fact that $x/\log(1/x) \leq 1$ whenever $0 < x \leq 1/2$.

Theorem 5.1. *For any discrete distribution X with support size c and for any $1/2 \geq \epsilon > 0$, there exists an estimator satisfying α-local differential privacy that estimates $H(X)$ within ϵ error using $n = O(c^2 \log \frac{1}{\delta} / (\epsilon^2 \alpha^2))$ samples with probability $1 - \delta$ when $\alpha \in (0, 1)$.*

This algorithm resulting from Theorem 5.1 can be used to privately estimate the entropy $H(X_i)$, mutual information $I(X_i; X_j)$, and conditional mutual information $I(X_i; X_j \mid X_k)$ of any variables X_i, X_j and X_k by using $O\left(\frac{\log \frac{1}{\delta}}{\alpha^2 \epsilon^2}\right)$ samples per estimate and $O(1)$ bits per sample, since each of these variables has constant support size, and both mutual information and conditional mutual information can be expressed in terms of entropies ((4) and (5) in Section 3). We call such an estimate $(\alpha, \epsilon, \delta)$-*good.*

5.2 Algorithm for General Trees

Note that the support size of X can be exponential in d. In the worst case, estimating the entropy of a distribution with support size k within constant error requires $\tilde{\Theta}(k)$ samples [19]. However the tree-structure of X can be exploited to significantly reduce the sample complexity. In their seminal paper, Chow and Liu [13] proved the identity

$$H(X) = \sum_{i=1}^{d} H(X_i) - \max_{T} \sum_{i=1}^{d} I(X_i; X_{\mathrm{pa}_T(i)}), \tag{6}$$

for any tree-structured random variable X, where the maximization is taken over all possible trees connecting the d variables.

Eq. (6) suggests a communication-efficient algorithm for estimating the entropy of X, which is known as the *Chow-Liu algorithm*: First, estimate each marginal entropy $H(X_i)$ and each mutual information $I(X_i; X_j)$. Next, compute a maximum spanning tree on the d variables, where the weight of each edge (X_i, X_j) is the estimate of the mutual information $I(X_i; X_j)$. Finally, plug all of the $H(X_i)$ estimates, the maximum spanning tree, and the corresponding $I(X_i; X_j)$ estimates, into Eq. (6).

The Chow-Liu algorithm requires $\Omega(d^2)$ samples, since it computes the mutual information between every pair of variables in order to compute a maximum spanning tree. However, estimating the right-hand side of Eq. (6) only requires estimating the *weight* of the maximum spanning tree, which is significantly easier than finding the tree itself. Algorithm 1 adapts a technique from [12] that estimates the weight of the maximum spanning tree of a graph in time that is sublinear in the number of edges in the graph. The basic idea is to select nodes of the graph at random and use breadth-first search to determine the size of each of their connected components if we were to drop edges that do not meet a weight threshold, short-circuiting the search when the size becomes too large. These quantities are combined to estimate the weight of the maximum spanning tree. In our case, an edge weight is a mutual information between a pair of variables, which we estimate from pair observations.

Theorem 5.2. *Algorithm 1 is α-locally differentially private and has $O(1)$ communication complexity. The number of samples requested by the server is $O\left(\frac{d \log(\frac{1}{\delta})}{\alpha^2 \epsilon^3}\right)$. Let $\hat{H}$ be the output of the algorithm. If X is tree-structured then $|\hat{H} - H(X)| \leq \epsilon d$ with probability $1 - \delta$.*

Algorithm 1 Shannon entropy estimation for tree-structured distribution

```
Let $M = \lceil \frac{2}{\epsilon} \rceil$ and $R = \lceil \frac{1}{\epsilon^2} \rceil$
for $m = 1, \ldots, M$ do
    for $r = 1, \ldots, R$ do
        Choose positive integer $Z$ randomly according to $\Pr[Z \geq z] = 1/z$.
        Choose $i^*$ uniformly at random from $[d]$.
        if $Z \geq \frac{2}{\epsilon}$ then $\gamma_{mr} \leftarrow 0$.
        else                                                    ▷ Breadth-first search
            Initialize queue to contain $i^*$ and a set $V = \{i^*\}$.
            while queue length is non-zero and shorter than $Z$ do
                Remove $i$ from front of queue and $V = V \cup \{i\}$.
                for $j = [d] \setminus V$ do
                    Server computes $(\alpha, \epsilon, \delta)$-good estimate $\hat{I}_{ij}$ of $I(X_i; X_j)$.
                    if $\hat{I}_{ij} \geq \epsilon m$ then add $j$ to back of queue.
            if queue length is zero then $\gamma_{mr} \leftarrow 0$ else $\gamma_{mr} \leftarrow 1$.
    $\hat{c}_m \leftarrow \frac{d}{R} \sum_{r=1}^{R} \gamma_{mr}$.
$\hat{W} \leftarrow \epsilon M d - \epsilon \sum_{m=1}^{M} \hat{c}_m$
Server computes $(\alpha, \epsilon, \delta)$-good estimate of each entropy in the first sum in Eq. (6).
Let $\hat{S}$ be the sum of the entropy estimates.
Return $\hat{H} = \hat{S} - \hat{W}$.
```

5.3 Algorithm for a Chain

Verma and Pearl [38] observed that if X is chain-structured with chain T then for any triplet (X_i, X_j, X_k), if X_k is on the unique path in T between X_i and X_j, then $I(X_i; X_j | X_k) = 0$. This observation alone does not help to recover the chain, since the conditional mutual information $I(X_i; X_j | X_k)$ can also be zero for X_i, X_j and X_k when X_k is not on the path between X_i and X_j in the chain T. Nevertheless, under the mild assumption that the mutual information $I(X_i, X_j)$ between every pair of variables is distinct, we can recover the chain T by estimating the conditional mutual information of triplets of variables.

The algorithm is similar to a sorting algorithm, such as "merge sort" which requires $O(d \log_2 d)$ pairwise comparisons over d items. While we cannot compare pairs explicitly like in a sorting problem, for any triplets (X_i, X_j, X_k), we can decide locally which "item" is between the other two: *i.e.*, $X_i \leftrightarrow X_j \leftrightarrow X_k$, $X_i \leftrightarrow X_k \leftrightarrow X_j$ or $X_k \leftrightarrow X_i \leftrightarrow X_j$ in the chain T, by estimating conditional mutual information. As the observation fro, Verma and Pearl implies that if X is chain-structured then one of $I(X_i; X_j | X_k)$ or $I(X_i; X_k | X_j)$ or $I(X_j; X_k | X_i)$ has to be zero (due to the observation from Verma and Pearl). This suggests our Algorithm 2, which inserts the variables in a chain one by one in a sequential manner. Algorithm 3 is called by Algorithm 2 as a subroutine which seeks to find the position where to insert.

Theorem 5.3. *Algorithm 2 is α-locally differentially private and has $O(1)$ communication complexity. The number of samples requested by the server is $O\left(\frac{d \log \frac{d}{\delta}}{\alpha^2 \epsilon^2}\right)$. Let $\hat{H}$ be the output of the algorithm. If X is chain-structured and $|I(X_i; X_j) - I(X_j; X_k)| \geq \epsilon$ then $|\hat{H} - H(X)| \leq \epsilon d$ with probability $1 - \delta$.*

5.4 Algorithm for a Star

If X is star-structured then recovering the star T is a matter of identifying the center of the star, which can be done by computing the mutual information between only $\tilde{\mathcal{O}}(d)$ pairs of variables. The algorithm picks a random marginal X_i and takes a "Prim's step", i.e., choosing the neighboring node (say X_k) that has the largest mutual information with X_i. The edge between X_i and X_k is in the maximal spanning tree, when we assume that each edges weight is different. Next, the algorithm estimates $\sum_{j \neq i} I(X_i, X_j)$ and $\sum_{j \neq k} I(X_k, X_j)$ to decide whether X_i or X_k is the center node of the star. Algorithm 4 presents the procedure, and Theorem 5.4 gives its sample complexity.

Algorithm 2 Shannon entropy estimation for chain-structured distribution

```
$S = [d]$, $C = \emptyset$, pick an arbitrary $i, j, k \in S$ and set $S = S \setminus \{i, j, k\}$.
Server computes $(\alpha, \epsilon, \delta)$-good estimates $\hat{I}(X_i; X_j \mid X_k)$, $\hat{I}(X_i; X_k \mid X_j)$ and $\hat{I}(X_k; X_j \mid X_i)$.
if $\hat{I}(X_i; X_j \mid X_k) > \epsilon$ then $\boldsymbol{x}_1 = (i, k, j)$
else if $\hat{I}(X_i; X_k \mid X_j) > \epsilon$ then $\boldsymbol{x}_1 = (i, j, k)$
else if $\hat{I}(X_k; X_j \mid X_i) > \epsilon$ then $\boldsymbol{x}_1 = (j, i, k)$
for $i \in (1, \ldots, d-3)$ do
    Pick item $j$ from $S$ and set $S = S \setminus \{j\}$ and set $r = x_{i,1}$ and $p = x_{i,i+2}$
    Server computes $(\alpha, \epsilon, \delta)$-good estimates $\hat{I}(X_j; X_p \mid X_o)$, $\hat{I}(X_r; X_j \mid X_p)$.
    if $\hat{I}(X_j; X_p \mid X_r) > \epsilon$ then $\boldsymbol{x}_{i+1} = (j, \boldsymbol{x}_i)$          ▷ Attach $X_j$ to the head of the chain
    else if $\hat{I}(X_o; X_j \mid X_p) > \epsilon$ then $\boldsymbol{x}_{i+1} = (\boldsymbol{x}_i, j)$      ▷ Attach $X_j$ to the tail of the chain
    else                                                  ▷ Insert $X_j$ into the chain defined by $\boldsymbol{x}_i$
        $\ell = $ TernarySearch$(\boldsymbol{x}_i, 1, i+2, j)$                        ▷ Defined in Algorithm 3
        $\boldsymbol{x}_{i+1} = (\boldsymbol{x}_i[1, \ldots, \ell], j, \boldsymbol{x}_i[\ell+1, \ldots, i+2])$
Create chain $T$ according to the order defined by $\boldsymbol{x}_{d-2}$.
Server computes $(\alpha, \epsilon, \delta)$-good estimate of each term in Eq. (6) using $T$ and returns their sum $\hat{H}$.
```

Algorithm 3 TernarySearch$(\boldsymbol{x}, \ell_l, \ell_h, j)$

```
if $\ell_l = \ell_h - 1$ then return $\ell_l$
Pick the median element $k = \lceil(\ell_h + \ell_l)\rceil$, and set $i = x_{\ell_l}$ and $o = x_{\ell_h}$
Server computes $(\alpha, \epsilon, \delta)$-good estimate $\hat{I}(X_i; X_k \mid X_j)$.
if $\hat{I}(X_i; X_k \mid X_j) > \epsilon$ then return TernarySearch$(\boldsymbol{x}, i, k, j)$
else return TernarySearch$(\boldsymbol{x}, k, o, j)$
```

Theorem 5.4. *Algorithm 4 is α-locally differentially private and has $O(1)$ communication complexity. The number of samples requested by the server is $O\left(\frac{d \log \frac{d}{\delta}}{\alpha^2 \epsilon^2}\right)$. Let $\hat{H}$ be the output of the algorithm. If X is star-structured and $|I(X_i; X_j) - I(X_j; X_k)| \geq \epsilon$ then $|\hat{H} - H(X)| \leq \epsilon d$ with probability $1 - \delta$.*

Algorithm 4 Shannon entropy estimation for star-structured distribution

```
Pick a $i \in [d]$ uniformly at random.
Server computes $(\alpha, \epsilon, \delta)$-good estimate $\hat{I}(X_i, X_j)$ for all $j \in [d] \setminus \{i\}$.
Find $k = \arg\max_{j \in [d] \setminus \{i\}} \hat{I}(X_i, X_j)$
Server computes $(\alpha, \epsilon, \delta)$-good estimate $\hat{I}(X_k, X_j)$.
if $\sum_j \hat{I}(X_i, X_j) > \sum_j \hat{I}(X_k, X_j)$ then let $T$ be a star with $X_i$ as center
else let $T$ be a star with $X_k$ as the center.
Server computes $(\alpha, \epsilon, \delta)$-good estimate of each term in Eq. (6) using $T$ and returns their sum $\hat{H}$.
```

5.5 Lower Bounds

We prove sample complexity lower bounds for estimating the Shannon entropy of a tree-structured distribution from pair observations. Our first lower bound focuses on the non-interactive case, when the algorithm must select all the pairs in advance. The second claim is more general, and holds for all sequentially interactive algorithms.

Theorem 5.5. *For any non-interactive algorithm that uses $o(d^2)$ pair observations to estimate Shannon entropy, there exists a tree-structured distribution over $\{0, 1\}^d$ such that the error of the algorithm is $\Omega(d)$ with constant probability.*

Theorem 5.6. *For any $\epsilon > 0$ and for any sequentially interactive algorithm that uses $o(d/\epsilon)$ pair observations to estimate the Shannon entropy, there exists a tree-structured distribution on $\{0, 1\}^d$ such that the error of the algorithm is $\Omega(\epsilon \cdot d)$ with constant probability.*

The lower bound given in Theorem 5.5 is based on Turán's theorem [9], which we use to show that for any algorithm with sub-quadratic sample complexity and for any constant $C \in (0, 1)$, there is a graph with d nodes containing $C \cdot d$-clique – when d is large enough – such that the algorithm does not observe any edge of that clique. This implies that the additive error of the algorithm is linear in d. The lower bound for sequentially interactive algorithms in Theorem 5.6 is based on a information theoretical approach. Interestingly, our construction of problem instances for which we applied Le Cam's theorem is fairly simple, since it contains d independent random variables in every case. Nevertheless, this lower bound shows that Algorithm 1 is optimal in d. Moreover, note that Theorem 5.6 also holds for non-interactive algorithms.

5.6 Comparison to Prior Work

To the best of our knowledge, the Chow-Liu algorithm is the only published method for estimating the entropy of a distribution that takes advantage of its tree structure. Since the algorithm is non-interactive, the lower bound in Theorem 5.5 shows that our algorithms have provably better sample complexity when the number of variables d is large (note that the dependence on d in each of Theorems 5.2, 5.3 and 5.4 is sub-quadratic). The Chow-Liu algorithm can also be used to estimate the distribution itself, not just its entropy, and it has recently been shown [8, 16] that the algorithm has optimal sample complexity when given full observations (*i.e.*, samples of the entire vector $(X_1, \ldots, X_d)$ and not just pairs or triplets of the variables). So the Chow-Liu algorithm is optimal for estimating a tree-structured distribution, but suboptimal for estimating the entropy of a tree-structured distribution. The root cause of this difference appears to be the fact that it is significantly easier to estimate the weight of the maximum spanning tree than finding the tree itself.

6 Estimating Gini and Collision Entropy

Algorithm 5 below estimates both the Gini entropy and collision entropy of a random variable X using samples from the distribution of X while observing only b bits per sample. Here, we do not require extra restrictions on the discrete distribution of X. In the algorithm, the server first partitions all users into pairs (we assume for simplicity that the number of users is even). The server then distributes a b-bit hash function to each user, along with a distinct cryptographic salt to each user pair. Each user then hashes their sample along with their salt, and returns the hash value to the server. The server computes entropy estimates based on the number of observed hash collisions across all pairs. In Algorithm 5 we let $\langle x, y \rangle$ denotes a binary string that encodes x, followed by a delimiter, and by y.

Algorithm 5 Gini and collision entropy estimation

```
Each user $i \in [n]$ draws sample $x_i$ independently from the distribution of $X$.
Server partitions the $n$ users into $\frac{n}{2}$ disjoint pairs.
Let $q_i \in \left[\frac{n}{2}\right]$ be the index of the pair containing user $i$.
Server transmits $q_i$ and hash function $h : \{0,1\}^* \mapsto \{0,1\}^b$ to each user $i$.
Each user $i$ generates a $b$-bit hash value $h_i = h(\langle q_i, x_i \rangle)$ for their sample $x_i$.
Each user $i$ lets $\hat{h}_i = h_i$ with probability $\lambda = \frac{e^\alpha}{2^b + e^\alpha}$ and otherwise draws $\hat{h}_i$ uniformly from $[2^b]$.
Server receives $\hat{h}_i$ from each user $i$.
If pair $q$ contains users $i$ and $j$ then let $c_q = \mathbf{1}[\hat{h}_i = \hat{h}_j]$ indicate whether a hash collision was
  observed for pair $q$.
Server computes $\bar{c} = \left(\frac{2}{(1-\lambda)n} \sum_q c_q\right) - \frac{\lambda}{2^b}$.
Server outputs $\hat{G} = \frac{2^b}{2^b-1}\bar{c} - \frac{1}{2^b-1}$ and $\hat{C} = -\log\left(1 - \hat{G}\right)$.
```

The analysis of Algorithm 5 is based on the observation that if X and X' are independent and identically distributed then the Gini entropy is equal to $1 - \Pr[X = X']$ and the collision entropy is equal to $-\log \Pr[X = X']$. If the server observed each sample directly then it could estimate $\Pr[X = X']$ using the collision frequency, *i.e.*, the fraction of sample pairs (x_i, x_j) such that $x_i = x_j$. However, the server only observes a b-bit hash of each sample. Among sample pairs in which there is a true collision, all of them also produce a hash collision. Among samples pairs in which there is not a true collision, about a $\frac{1}{2^b}$ fraction of them produce a hash collision. Therefore the true collision frequency can be estimated using an appropriately bias-corrected hash collision frequency, and the server uses this estimate to approximate the Gini entropy and collision entropy.

The analysis of Algorithm 5 is given in Theorem 6.1 below. As is customary, for the analysis we assume that the hash function h is constructed by assigning each element of its domain to a uniform random element of its range [7]. See the Appendix for the proof of the theorem.

Theorem 6.1. *Algorithm 5 is α-locally differentially private and has $\tilde{O}(b)$ communication complexity and $\tilde{O}(b)$ space complexity. Let $\hat{G}$ and $\hat{C}$ be the outputs of the algorithm. Let $\alpha, \epsilon, \delta \in (0, 1)$. If $n = \Omega\left(\frac{b^2 \max\{1-G(X), 2^{-b}\} \log\frac{1}{\delta}}{\alpha^2\epsilon^2}\right)$ then $|\hat{G} - G(X)| \leq \epsilon$ with probability at least $1 - \delta$. Also, if X has support size k and $n = \Omega\left(\frac{b^2k^2 \log\frac{1}{\delta}}{\alpha^2\epsilon^2 \min\{k, 2^b\}}\right)$ then $|\hat{C} - C(X)| \leq \epsilon$ with probability at least $1 - \delta$.*

6.1 Comparison to Prior Work

Recall that Gini entropy is one minus the second frequency moment (up to a constant). Local differentially private algorithms for estimating frequency moments were recently studied in [11]. Letting $b = 1$ in Algorithm 5 yields a sample complexity of $\tilde{O}(1/\alpha^2\epsilon^2)$, which matches the sample complexity of the sequentially interactive algorithm for estimating the second frequency moment from [11]. However our algorithm is non-interactive, which is a much weaker communication model. Also it only uses 1 bit per sample and $\tilde{O}(1)$ space, while the previous algorithm uses $\Omega(k)$ bits per sample and $\Omega(k)$ space, where k is the support size of the distribution. The authors in [11] asked whether there is a non-interactive algorithm for privately estimating frequency moments with a sample complexity that is independent of the distribution's support size. Here we affirmatively answer this open question for the second frequency moment.

The best known algorithm for estimating collision entropy using $\tilde{O}(1)$ space is due to [35]. The sample complexity of their algorithm is $\tilde{O}\left(k/\epsilon^2\right)$ and its communication complexity is $O(\log k)$ communication per user, where k is the support size of the distribution. Letting $b = \log k$ and $\alpha = O(1)$ in Algorithm 5 recovers these results (up to logarithmic factors), and using smaller values for b or α generalizes the previous algorithm to the private and communication-efficient setting. Also, it was shown in [15] that (conditioned on a plausible conjecture) any algorithm that estimates collision entropy to within $O(1)$ error using $O(1)$ space requires $\Omega(k)$ samples. Therefore our algorithm is likely to be Pareto optimal with respect to the sample complexity-space complexity trade-off.

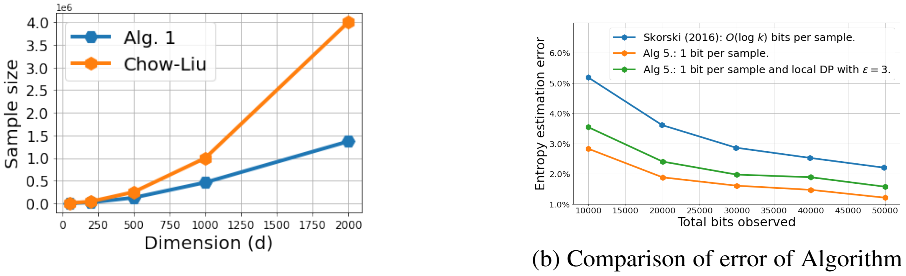
7 Experiments

In this section we present two sets of experiments to support our theoretical findings. First, we justify that Algorithm 1 is indeed amenable to estimate the Shannon entropy of tree-based distributions with linear sample complexity in d. Thus it has a superior sample complexity comparing to the state-of-the-art non-interactive method [13, 8] which has a quadratic sample complexity in d. The sample complexity is defined here in terms of the number of observation from pairs of marginals. In the second set of experiments, we test Algorithm 5 to estimate the collision entropy of discrete distributions, and we compare its performance to the best known communication efficient, non-private algorithm, to our best knowledge. We refer to this algorithm as Skorski's algorithm [35].

Estimating Shannon entropy: To estimate the Shannon entropy as it is given in (6), the marginal entropy values and the mutual information between certain or all pairs of marginals have to be estimated. More concretely, Chow-Liu algorithm estimates the mutual information between all pairs of marginals which results in quadratic sample complexity, whereas Algorithm 1 estimates the mutual information only for linear fraction of pairs. What is common in these two algorithms is that they both estimates the marginal entropy values by sampling the each marginals independently from the mutual information estimations. In addition to this, both algorithm estimates the mutual information between pairs of marginals with the same ϵ additive error and with the same privacy budget α. Therefore it is fair to compare the performance of these two algorithms in terms of number pairs for which the mutual information is estimated by them. For Chow-Liu algorithm this is d^2 whereas Algorithm 1 is a randomized algorithm, thus we evaluate it over 100 repetitions and report the average.

In this experiments, we used tree-structured multivariate distributions over $\{0, 1\}^d$. The tree structure is picked randomly by taking the maximum spanning tree in a full graph with edge weights that obey standard normal distribution. Then we chose parameters for each marginals uniformly at random from

$[0,1]$. The tree structure is achieved by inducing dependence between marginal distribution while preserving the marginals. More concretely, for two marginals X_i and X_j with parameters p_i and p_j, we set $P((X_i, X_j) = (0,1)) = (1-p_i) \cdot p_j + d$, $P((X_i, X_j) = (0,0)) = (1-p_i) \cdot (1-p_j) - d$, $P((X_i, X_j) = (1,0)) = p_i \cdot (1-p_j) - d$ and $P((X_i, X_j) = (1,1)) = p_i \cdot p_j + d$ where selected d uniformly at random so as each probability stays positive.



(a) Comparison of sample complexity of Algorithm 1 and Chow-Liu for estimating Shannon entropy as given in (6) for tree structured distribution

(b) Comparison of error of Algorithm 5 and Skorski's algorithm [35] for estimating collision entropy for exponential distribution with domain size $k = 1000$.

The results are shown in Figure 1a. One can clearly seen that the sample size, i.e. number of mutual information estimate, is close to linear however Chow-Liu algorithm requires a higher sample size. Note that sample size means only the pairs of marginals for which the algorithms requires to estimate the mutual information.

Estimating collision entropy: In this set of experiment, we focus on estimating collision entropy. We drew samples from a discrete exponential distribution $p_i \propto e^{-i}$ with support size $k = 1000$. We used the previously best-known algorithm [35], which requires $O(\log k)$ bits per sample and is not private, to estimate the collision entropy of the distribution. We also used our algorithm, which requires only 1 bit per sample and can be made differentially private. The results is viewed in Figure 1b. Our experiment shows that the previous algorithm has 5% estimation error after observing 10000 bits, while our algorithm has less than 3.5% estimation error. Thus our algorithm has lower error for the same communication cost, and it is also (local) differentially private.

8 Conclusion and future work

Estimating entropy is of importance in many practical applications. In this paper, we studied three widely used entropy measures: Shannon, Gini and collision entropy. We described estimation algorithms for each entropy that require minimal communication and satisfy local differential privacy. Our sequentially interactive algorithm for estimating the Shannon entropy of high-dimensional tree-structured distributions observes only two of these dimensions per sample, and has a sample complexity $O(d/\epsilon^3)$. Our approach relies on the celebrated Chow-Liu approximation [13]. It provides an improvement of the sample complexity of the original non-interactive Chow-Liu algorithm whose sample complexity is $\Omega(d^2)$. We also identified some special cases when the underlying graphical model of the joint distribution is either a chain or star graph. In these cases, the proposed algorithms have a sample complexity of $\tilde{O}(d \log d/\epsilon^2)$. Our algorithm for estimating the Gini and collision entropy estimation also improved on the state-of-the-art, either by matching the sample complexity of previous work but in a weaker communication model and with significantly better communication complexity, or by generalizing the best known algorithm to the private and communication-efficient setting. Lastly, we demonstrated the versatility of our methods on synthetic data, and showed that the proposed methods for Shannon entropy and collision entropy are superior when comparing them to the performance to the state-of-the-art methods.

A natural extension of our work on Shannon entropy estimation is to consider higher-order correlations in the Chow-Liu decomposition [25]. In this case, discovering the underlying structure of the joint distribution is already computationally challenging, unlike in the second order case when it reduces to a maximum spanning tree problem. However, efficiently estimating the entropy of the resulting distribution might still be possible.

References

[1] Jayadev Acharya, Clement Canonne, Cody Freitag, and Himanshu Tyagi. Test without trust: Optimal locally private distribution testing. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 2067–2076. PMLR, 2019.

[2] Jayadev Acharya, Clément L Canonne, and Himanshu Tyagi. Inference under information constraints: Lower bounds from chi-square contraction. In *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 3–17. PMLR, 2019.

[3] Jayadev Acharya, Gautam Kamath, Ziteng Sun, and Huanyu Zhang. Inspectre: Privately estimating the unseen. In *International Conference on Machine Learning*, pages 30–39. PMLR, 2018.

[4] Jayadev Acharya and Ziteng Sun. Communication complexity in locally private distribution estimation and heavy hitters. In *International Conference on Machine Learning*, pages 51–60. PMLR, 2019.

[5] Jayadev Acharya, Ziteng Sun, and Huanyu Zhang. Hadamard response: Estimating distributions privately, efficiently, and with little communication. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1120–1129. PMLR, 2019.

[6] John C. Baez, Tobias Fritz, and Tom Leinster. A characterization of entropy in terms of information loss. *Entropy*, 13(11):1945–1957, 2011.

[7] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pages 62–73, 1993.

[8] Arnab Bhattacharyya, Sutanu Gayen, Eric Price, and NV Vinodchandran. Near-optimal learning of tree-structured distributions by chow-liu. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 147–160, 2021.

[9] Béla Bollobás. *Modern Graph Theory*. Graduate Texts in Mathematics 184. 1998.

[10] Tai-Danae Bradley. Entropy as a topological operad derivation. *Entropy*, 23(9), 2021.

[11] Cristina Butucea and Yann ISSARTEL. Locally differentially private estimation of functionals of discrete distributions. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 24753–24764. Curran Associates, Inc., 2021.

[12] Bernard Chazelle, Ronitt Rubinfeld, and Luca Trevisan. Approximating the minimum spanning tree weight in sublinear time. *SIAM Journal on computing*, 34(6):1370–1379, 2005.

[13] C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.

[14] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, New York, NY, USA, 2006.

[15] Michael Crouch, Andrew McGregor, Gregory Valiant, and David P Woodruff. Stochastic streams: Sample complexity vs. space complexity. In *24th Annual European Symposium on Algorithms (ESA 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[16] Constantinos Daskalakis and Qinxuan Pan. Tree-structured ising models can be learned efficiently. *arXiv e-prints*, pages arXiv–2010, 2020.

[17] John Duchi and Ryan Rogers. Lower bounds for locally private estimation via communication complexity. In *Conference on Learning Theory*, pages 1161–1191. PMLR, 2019.

[18] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

[19] Kazuto Fukuchi and Jun Sakuma. Minimax optimal estimators for additive scalar functionals of discrete distributions. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 2103–2107. IEEE, 2017.

[20] C. Gini. *Variabilitá e Mutuabilitá. Contributo allo Studio delle Distribuzioni e delle Relazioni Statistiche*. Bologna: C. Cuppini., 1912.

[21] Yanjun Han, Ayfer Özgür, and Tsachy Weissman. Geometric lower bounds for distributed parameter estimation under communication constraints. In *Conference On Learning Theory*, pages 3163–3188. PMLR, 2018.

[22] Oc Herfindahl. Concentration in the us steel industry. 1950.

[23] Matthew Joseph, Jieming Mao, Seth Neel, and Aaron Roth. The role of interactivity in local differential privacy. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 94–105. IEEE, 2019.

[24] Matthew Joseph, Jieming Mao, Seth Neel, and Aaron Roth. The role of interactivity in local differential privacy. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 94–105. IEEE Computer Society, 2019.

[25] Edith Kovács and Tamás Szántai. *On the Approximation of a Discrete Multivariate Probability Distribution Using the New Concept of t-Cherry Junction Tree*, pages 39–56. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[26] Markku Laakso and Rein Taagepera. “effective” number of parties: A measure with application to west europe. *Comparative Political Studies*, 12(1):3–27, 1979.

[27] Tom Leinster. *Entropy and Diversity: The Axiomatic Approach*. Cambridge University Press, 2021.

[28] Abolfazl S Motahari, Guy Bresler, and NC David. Information theory of DNA shotgun sequencing. *IEEE Transactions on Information Theory*, 59(10):6273–6289, 2013.

[29] Laura Elena Raileanu and Kilian Stoffel. Theoretical comparison between the Gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence*, 41(1):77–93, 2004.

[30] Florencia Rao. `https://www.secureauth.com/blog/the-art-of-war-of-browser-fingerprinting/`, 2021.

[31] Alfréd Rényi. On measures of information and entropy. In *Proceedings of the fourth Berkeley Symposium on Mathematics, Statistics and Probability*, pages 547–561, 1960.

[32] C.E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27:379–423, 1948.

[33] Edward H Simpson. Measurement of diversity. *nature*, 163(4148):688–688, 1949.

[34] Yakov G. Sinai. On the notion of entropy of a dynamical system. 2010.

[35] Maciej Skorski. Evaluating entropy for true random number generators: Efficient, robust and provably secure. In *International Conference on Information Security and Cryptology*, pages 526–541, 03 2017.

[36] Ananda Theertha Suresh, X Yu Felix, Sanjiv Kumar, and H Brendan McMahan. Distributed mean estimation with limited communication. In *International Conference on Machine Learning*, pages 3329–3337. PMLR, 2017.

[37] Constantino Tsallis. *Introduction to Nonextensive Statistical Mechanics*. Springer New York, NY, Greece, 2009.

[38] Thomas Verma and Judea Pearl. Equivalence and synthesis of causal models. In *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, UAI ’90, page 255–270, USA, 1990. Elsevier Science Inc.

[39] Yuchen Zhang, John C Duchi, Michael I Jordan, and Martin J Wainwright. Information-theoretic lower bounds for distributed statistical estimation with communication constraints. In *NIPS*, pages 2328–2336. Citeseer, 2013.

Checklist

1. For all authors...
   - (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   - (b) Did you describe the limitations of your work? [Yes] There is a gap between upper and lower bounds.
   - (c) Did you discuss any potential negative societal impacts of your work? No social negative impact.
   - (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   - (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   - (b) Did you include complete proofs of all theoretical results? [Yes] In the appendix.
3. If you ran experiments...
   - (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? No experiments.
   - (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? No experiments.
   - (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? No experiments.
   - (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? No experiments.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   - (a) If your work uses existing assets, did you cite the creators? We used results published in peer reviewed venues and we cited these results properly.
   - (b) Did you mention the license of the assets? No licence is needed.
   - (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   - (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   - (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   - (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   - (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   - (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Proof of Theorem 5.1

Proof. [4, 5] showed that any discrete distribution can be learnt in total variation distance based on $O(c^2 \log 1/\delta/(\epsilon^2\alpha^2)$ when $\alpha < 1$. This result can be plug-in into Theorem 17.3.3 of [14]. □

B Proof of Theorem 5.2

The result follows from simply combining the algorithm from [12] for estimating the weight of the maximum spanning tree in sublinear time, which assumes that each edge weight can be computed in $O(1)$ time, with Theorem 5.1, which gives the number of samples needed to privately estimate an edge weight (*i.e.*, the mutual information between two variables). The main technical complication we must overcome is that the result from [12] assumes that the edge weights are integers with a bounded ratio, while we instead discretize the edge weights with resolution ϵ. To complete the proof, we need the Lemma B.1 below, which concerns a graph G whose edge weights are multiples of ϵ. The lemma relates the weight of the maximum spanning tree of G to the number of connected components in various subgraphs of G. This lemma replaces Claim 5 from [12].

Let G be a connected graph with n vertices and edge weights belonging to the set

$$\{\epsilon, 2\epsilon, \ldots, w\epsilon\},$$

where $\epsilon > 0$ and w is a positive integer.

For each $i \in \{1, \ldots, w\}$ let G_i be the subgraph of G containing the same vertices as G and only those edges of G whose weight is at least $i\epsilon$. Thus $G_1 = G$. Let c_i be the number of connected components in G_i. Let M be the weight of a maximum spanning tree of G.

Lemma B.1. $M = \epsilon wn - \epsilon \sum_{i=1}^{w} c_i$.

Proof. For each $i \in \{1, \ldots, w\}$ let γ_i be the number of edges with weight $i\epsilon$ in a maximum spanning tree of G. We have

$$\sum_{i<\ell} \gamma_i = c_\ell - 1$$

for any $\ell \in \{1, \ldots, w\}$, where the empty sum is defined to be zero. This equality can be established by considering Kruskal's greedy algorithm for constructing a maximum spanning tree, which adds edges in decreasing order of weight as long as they do not induce a cycle. Since G_ℓ has c_ℓ connected components, and all the edges in G_ℓ are heavier than all the edges not in G_ℓ, the greedy algorithm must first connect the vertices within each component of G_ℓ and then use exactly $c_\ell - 1$ edges not in G_ℓ to connect the components to each other. Thus we have

$$\begin{aligned}
M &= \sum_{i=1}^{w} i\epsilon\gamma_i \\
&= \epsilon\left(\sum_{i\geq 1}\gamma_i + \sum_{i\geq 2}\gamma_i + \cdots + \sum_{i\geq w}\gamma_i\right) \\
&= \epsilon\left(n - 1 - \sum_{i<1}\gamma_i + n - 1 - \sum_{i<2}\gamma_i + \cdots + n - 1 - \sum_{i<w}\gamma_i\right) \\
&= \epsilon\left(w(n-1) - \sum_{i=1}^{w}(c_i - 1)\right) \\
&= \epsilon wn - \epsilon\sum_{i=1}^{w} c_i
\end{aligned}$$

□

C Proof of Theorem 5.3

We start by recalling a lemma that applies to tree-structured distributions.

Lemma C.1. *[38] Let $\mathbf{X} = (X_1, \ldots, X_d)$ be tree decomposable with tree T. Then for any triplets i, j and k, if k is on the unique path in T between i and j, then*

$$I(X_i; X_j | X_k) = 0 \ .$$

Next we show that the output Algorithm 2 is correct with high probability. We make use of the Conditional Mutual Information Tester of [8]. This testing algorithm consists of estimating the CMI using the plug-in estimator and then applying a ϵ threshold on the estimate, i.e. if the estimate is smaller than ϵ then accept, otherwise reject. The sample complexity of Conditional Mutual Information Tester is $O\left(\frac{|\Sigma|^3}{\epsilon} \log \frac{d|\Sigma|}{\delta} \log \frac{|\Sigma| \log d/\delta}{\epsilon}\right)$ according to Theorem 1.3 of [8], thus if we apply this tester adjusted confidence parameter, i.e. $\delta/d \log_3 d$ then the union bound implies that the output of all test is correct with probability at least $1 - \delta$.

Next, note that for any any triplet X_i, X_j, X_k such that X_i is between X_j and X_k in the chain, it holds that

$$I(X_j; X_i) - I(X_j; X_k) = \underbrace{I(X_j; X_i | X_k)}_{>\epsilon} - \underbrace{I(X_j; X_k | X_i)}_{=0 \text{ due to Lemma C.1}}$$

because of the edges are different with a margin of ϵ. Same argument implies that $I(X_k; X_i | X_j) > \epsilon$. Thus, Algorithm 2 divides the nodes correctly in Line 4-10 which along with the testers' correctness with high probability implies the correctness of the algorithm.

The sample complexity of Conditional Mutual Information Tester is $O\left(\frac{|\Sigma|^3}{\epsilon} \log \frac{d|\Sigma|}{\delta} \log \frac{|\Sigma| \log d/\delta}{\epsilon}\right)$ according to Theorem 1.3 of [8], we only upper bound the number of tests for deciding whether $I(X_i; X_j \mid X_k) > \epsilon$ that is carried out by Algorithm 2. It is indeed $3 \sum_{i=3}^{d} \log_3 i \in O(d \log_3 d)$ which concludes the proof.

D Proof of Theorem 5.4

First note that the Prim step in Line of Algorithm 4 indeed finds the edge that is in the maximum spanning tree due to the assumption $|I(X_i; X_j) - I(X_j; X_k)| \geq \epsilon$. Say the edge which is found by Prim step is between X_i and X_k. What remained is to decide whether X_i or X_k is the center of the graph which can done by comparing $\sum_j \hat{I}(X_i, X_j) > \sum_j \hat{I}(X_k, X_j)$. According to Theorem 5.1, entropy can estimated with ϵ error with α-locally differential private guaranty using $O(c^2 \log \frac{1}{\delta} / (\epsilon^2 \alpha^2))$ samples. Note that Algorithm 4 estimates $2d$ mutual information which requires $4d$ marginal and $2d$ pairwise marginal entropy estimation which justifies the scaling of the confidence parameter, thus the algorithm is correct if the confidence parameter is set to $\delta/6d$ due to the union bound and sample complexity is accordingly $O(\frac{6dc^2}{\epsilon^2 \alpha^2} \log \frac{6d}{\delta})$.

E Proof of Theorem 5.5

Assume a deterministic algorithm $\mathcal{A}$ which takes sub-quadratic samples from $\mathbf{X}$ and estimates $H(\mathbf{X})$. In addition, we assume that its sample complexity is $o(d^2)$. Thus $\forall C (> 0)$ there exists d_0 such that for any $d > d_0$ it holds the sample complexity of the algorithm is $< Cd^2$ which implies that, if d is large enough the algorithm needs $\leq d^{2-\kappa}$. In addition, we can pick d so as $d^{-\kappa} < \kappa$ which implies $d^2 - d^{2-\kappa} > (1 - \kappa) d^2$ thus any deterministic algorithm which takes sub-quadratic sample size never observes $(1 - \kappa) d^2$ edges for $\forall \kappa$ when d is large enough.

Let us recall that Turán's theorem

Theorem E.1 ([9])**.** *Let $G(V, E)$ be a graph with graph vertices V and graph edges E on d graph vertices without a $(\ell + 1)$-clique. Then*

$$t(d, \ell) \leq \frac{(\ell - 1) d^2}{2\ell},$$

where $t(n, k)$ is the edge count.

In addition, Turán's graph $G_T(d, \ell)$ [9] is defined as the unique graph without a $(\ell+1)$-clique having the maximum possible number of graph edges which is

$$t(d, \ell) = \left\lfloor \frac{(\ell-1)d^2}{2\ell} \right\rfloor$$

Thus for any $d > 0$ there exists a graph $G = (V, E)$ such that $|V| = d$ and $|E| > t(d, k)$ and contains a $(k+1)$-clique. This implies that for any algorithm $\mathcal{A}$, if it does not observes at least $t(d, k)$ edges, then there exists a $k+1$-clique of which any edges is never observed by algorithm $\mathcal{A}$. Now apply Turan's result with $\ell = d/2$ which implies that

$$t(d, d/2) = \left\lfloor (1/2 - 1/d)d^2 \right\rfloor$$

Thus for any $\kappa \in (0, 1/2)$ and any algorithm with sample complexity $o(d^2)$, if d is large enough, then there will be a $\Theta(d/2)$-clique for which the algorithm does not observe any edge within this clique.

Finally we can easily construct two $d/2$-dimensional problem instances, denoted by S and S' for which the joint entropy differs by $\Omega(d)$: let us take $d/2$ Bernoulli with parameter $1/2$ and take the copy of the same Bernoulli $d/2$ times. The entropy is $d/2$ and 1 for these two joint distributions. This also implies that for any deterministic algorithm we can construct two problem instances which contains S and S' so as they are independent from the rest of the marginals, and the algorithm does not observe any sample from them, thus it cannot achieve $o(d)$ additive error.

F Proof of Theorem 5.6

Let $\hat{\theta}_n = \hat{\theta}(x_1, \ldots, x_n)$ such that $\hat{\theta}_n : (\Sigma^d)^n \mapsto \mathbb{R}$ be an estimator using n samples.

Theorem F.1. *[Le Cam's theorem] Let $\mathcal{P}$ be a set of distributions. Then, for any pair of distributions $P_0, P_1 \in \mathcal{P}$, we have*

$$\inf_{\hat{\theta}} \max_{P \in \mathcal{P}} \mathbb{E}_P \left[d(\hat{\theta}_n(P), \theta(P)) \right] \geq \frac{d(\theta(P_0), \theta(P_1))}{8} e^{-n d_{KL}(P_0, P_1)},$$

where $\theta(P)$ is a parameter taking values in a metric space with metric d, and $\hat{\theta}_n$ is the estimator of θ based on n samples.

Let us consider two Bernoulli distributions P_0 and P_1 with parameters $p_0 = 1/2$ and $p_1 = 1/2 - \epsilon$, where $\epsilon \in (0, 1/2)$. The entropy of random variables X_0 and X_1 distributed according to P_0 and P_1 are $H(X_0) = 1$ and

$$H(X_1) = -\left(\frac{1}{2} - \epsilon\right) \log_2 \left(\frac{1}{2} - \epsilon\right) - \left(\frac{1}{2} + \epsilon\right) \log_2 \left(\frac{1}{2} + \epsilon\right).$$

Thus,

$$\begin{aligned} |H(X_0) - H(X_1)| = H(X_0) - H(X_1) &= \left(\epsilon + \frac{1}{2}\right) \log_2(1 + 2\epsilon) - \left(\epsilon - \frac{1}{2}\right) \log_2(1 - 2\epsilon) \\ &\geq \left(\epsilon + \frac{1}{2}\right) \frac{2\epsilon}{1 + 2\epsilon} - \left(\frac{1}{2} - \epsilon\right) 2\epsilon \\ &\geq 2\epsilon^2 \end{aligned}$$

where we used that $\log(1 - 2\epsilon) \leq -\epsilon$ for $0 < \epsilon < 1$ and $\frac{\epsilon}{1+\epsilon} \leq \log(1 + \epsilon)$ for $\epsilon > -1$. The KL divergence can be upper bounded as

$$d_{\mathrm{KL}}(P_0, P_1) = -\frac{1}{2} \log_2(1 - 4\epsilon^2) \leq 2\epsilon^2.$$

We can now apply the Le Cam's theorem for the set of Bernoulli distributions with metric d being the ℓ_1-norm as

$$\inf_{\hat{\theta}} \max_{P \in \mathcal{P}} \mathbb{E}_P \left[|\hat{\theta}_n(P) - H(P)| \right] \geq \frac{d(\theta(P_0), \theta(P_1))}{8} e^{-n d_{\mathrm{KL}}(P_0, P_1)} \geq \frac{\epsilon^2}{4} e^{-2n\epsilon^2}$$

Using this result with $\epsilon' = \sqrt{\epsilon}$, the following sample complexity can be obtained for estimating Shannon entropy.

Corollary F.2. *For any $\hat{\theta}_n$ such that $n \in o(1/\epsilon)$, there exists a Bernoulli distribution P for which*

$$\mathbb{E}_P\left[|\hat{\theta}_n(P) - H(P)|\right] \geq C \cdot \epsilon,$$

with $C > 0$ that does not depend on ϵ.

First of all, notice that some bound on error $r(\delta)$, either lower or upper, that holds with probability $1-\delta$, translates into the bound $r(\delta)+\delta$ on the expected error in a straightforward manner. Thus the lower bound presented in Corollary F.2 also implies that there is no high probability estimator for entropy with $o(1/\epsilon)$ sample complexity for discrete distributions. This can be used to lower bound of the entropy estimator for joint distribution as follows. Let $\mathcal{B} = \{\mathbf{b} = (b_1, \dots, b_d) : b_j \in \{0,1\}\}$ are the vertices of the d dimensional hypercube, and let us define a set of d-dimensional distribution $\mathcal{P}_\mathbf{b}$ indexed by the element of $\mathcal{B}$. Each $P_\mathbf{b} \in \mathcal{P}$ contains $X_0 \sim$ Bern$(1/2)$ if $b_i = 0$ and $X_1 \sim$ Bern$(1/2 - \epsilon)$ if $b_i = 1$, i.e.

$$P_\mathbf{b} = X_{b_1} \oplus \cdots \oplus X_{b_d}$$

and

$$\mathcal{P} = \left\{P_\mathbf{b} : \mathbf{b} \in \{0,1\}^d\right\} \ .$$

It is clear that $\mathcal{P}$ is a subset of the tree-structured distributions and each distribution contains d independent Bernoulli random variables, thus

$$H(P_\mathbf{b}) = \sum_{i=1}^{d} H(X_{b_i})$$

Therefore any estimator that achieves at most $\epsilon \cdot d$ additive error for $H(P_\mathbf{b})$ has to estimate each individual Bernoulli distribution with at most ϵ error. The sample complexity of any estimator of $H(P_\mathbf{b})$ with an additive error $O(\epsilon d)$ is $\Omega(d/\epsilon)$.

G Proof of Theorems 6.1

Algorithm 5 clearly has $\tilde{O}(b)$ communication complexity and $\tilde{O}(b)$ space complexity, since it only has to maintain a counter of collisions between b-bit hashes. Each user replaces their hash with a random hash with probability λ, and therefore the algorithm is α-local differentially private, since $\log\left(\frac{\lambda}{(1-\lambda)/2^b}\right) = \log\left(\frac{\lambda 2^b}{1-\lambda}\right) = \alpha$ where we used $\lambda = \frac{e^\alpha}{e^\alpha + 2^b}$. Before proving the sample complexity we will first prove the following result.

Lemma G.1. *Let $\hat{G}$ be output by Algorithm 5. Let $\epsilon, \delta \in (0,1)$. If*

$$n \geq \frac{6b^2 \log \frac{2}{\delta}}{\alpha^2 \epsilon^2 \left(\left(1 - \frac{1}{2^b}\right) G(X) + \frac{1}{2^b}\right)}$$

then $|\hat{G} - G(X)| \leq \epsilon\left(G(X) + \frac{1}{2^b - 1}\right)$ with probability at least $1 - \delta$.

Proof. Recall that if X and X' are independent and identically distributed then

$$G(X) = 1 - \Pr[X = X'].$$

We will calculate the expected value of each c_q. Suppose pair q contains samples x_i and x_j. If $x_i = x_j$ then $c_q = 1$ with probability $1 - \lambda + \frac{\lambda}{2^b}$, and otherwise $c_q = 1$ with probability $\frac{1}{2^b}$. Thus

$$\begin{aligned}
\mathrm{E}[c_q] &= \Pr[c_q = 1 \mid x_i = x_j]\Pr[x_i = x_j] + \Pr[c_q = 1 \mid x_i \neq x_j]\Pr[x_i \neq x_j] \\
&= \left(1 - \lambda + \frac{\lambda}{2^b}\right)\Pr[x_i = x_j] + \frac{1}{2^b}\Pr[x_i \neq x_j] \\
&= \left(1 - \lambda + \frac{\lambda}{2^b}\right)\Pr[x_i = x_j] + \frac{1}{2^b}(1 - \Pr[x_i = x_j]) \\
&= \left(1 - \lambda + \frac{\lambda}{2^b} - \frac{1}{2^b}\right)\Pr[x_i = x_j] + \frac{1}{2^b}
\end{aligned}$$

$$= \left(1 - \lambda + \frac{\lambda}{2^b} - \frac{1}{2^b}\right)(1 - G(X)) + \frac{1}{2^b}$$

where the last line follows because x_i and x_j independent samples from the distribution of X.

Recall that by the Chernoff bound if $z_1, \ldots, z_m$ are independent random variables such that $z_i \in \{0,1\}$ then for all $\epsilon \in (0,1)$ the average $\bar{z} = (z_1 + \cdots z_m)/m$ satisfies

$$\Pr\left[\bar{z} \geq (1+\epsilon)\,\mathrm{E}[\bar{z}]\right] \leq \exp\left(-\frac{\epsilon^2 m}{3}\,\mathrm{E}[\bar{z}]\right), \text{ and}$$
$$\Pr\left[\bar{z} \leq (1-\epsilon)\,\mathrm{E}[\bar{z}]\right] \leq \exp\left(-\frac{\epsilon^2 m}{3}\,\mathrm{E}[\bar{z}]\right).$$

The c_q's are independent random variables because each c_q is defined using a distinct pair of samples and distinct pair index. Also, each $c_q \in \{0,1\}$. Note that we proved above that the average of the c_q's is $\frac{2}{n}\sum_q \mathrm{E}[c_q] = \left(1 - \lambda + \frac{\lambda}{2^b} - \frac{1}{2^b}\right)(1 - G(X)) + \frac{1}{2^b}$. Therefore

$$\begin{aligned}
\Pr\left[\hat{G} \geq G(X) + \epsilon\left(G(X) + \frac{1}{2^b - 1}\right)\right] &= \Pr\left[\frac{2^b}{2^b-1}\bar{c} - \frac{1}{2^b-1} \geq G(X) + \epsilon\left(G(X) + \frac{1}{2^b - 1}\right)\right] \\
&= \Pr\left[\bar{c} \geq (1+\epsilon)\left(\left(1 - \frac{1}{2^b}\right)G(X) + \frac{1}{2^b}\right)\right] \\
&= \Pr\left[\bar{c} \geq (1+\epsilon)\,\mathrm{E}[\bar{c}]\right] \\
&\leq \exp\left(-\frac{\epsilon^2 n}{6}\,\mathrm{E}[\bar{c}]\right) \\
&= \exp\left(-\frac{\alpha^2\epsilon^2 n}{6b^2}\left(\left(1 - \frac{1}{2^b}\right)(1 - G(X)) + \frac{1}{2^b}\right)\right)
\end{aligned}$$

where the inequality follows from the Chernoff upper bound. By a very similar calculation

$$\begin{aligned}
\Pr\left[\hat{G} \leq G(X) - \epsilon\left(G(X) + \frac{1}{2^b - 1}\right)\right] &= \Pr\left[\frac{2^b}{2^b-1}\bar{c} - \frac{1}{2^b-1} \leq G(X) - \epsilon\left(G(X) + \frac{1}{2^b - 1}\right)\right] \\
&= \Pr\left[\bar{c} \leq (1-\epsilon)\left(\left(1 - \frac{1}{2^b}\right)G(X) + \frac{1}{2^b}\right)\right] \\
&= \Pr\left[\bar{c} \leq (1-\epsilon)\,\mathrm{E}[\bar{c}]\right] \\
&\leq \exp\left(-\frac{\epsilon^2 n}{6}\,\mathrm{E}[\bar{c}]\right) \\
&= \exp\left(-\frac{\alpha^2\epsilon^2 n}{6b^2}\left(\left(1 - \frac{1}{2^b}\right)(1 - G(X)) + \frac{1}{2^b}\right)\right)
\end{aligned}$$

where the inequality follows from the Chernoff lower bound. Combining the above we have

$$\Pr\left[\left|\hat{G} - G(X)\right| \geq \epsilon\left(1 - G(X) + \frac{1}{2^b - 1}\right)\right] \leq 2\exp\left(-\frac{\alpha^2\epsilon^2 n}{6b^2}\left(\left(1 - \frac{1}{2^b}\right)(1 - G(X)) + \frac{1}{2^b}\right)\right)$$

and rearranging proves the lemma. □

Now the first sample complexity bound in the theorem follows immediately from Lemma G.1. As for the second sample complexity bound, since $C(X) = -\log(1 - G(X))$ we have

$$\begin{aligned}
\left|C(X) - \hat{C}\right| = \left|\log(1 - \hat{G}) - \log(1 - G(X))\right| = \left|\log\frac{1 - \hat{G}}{1 - G(X)}\right| \\
\leq \log\left(1 + \frac{\left|\hat{G} - G(X)\right|}{1 - G(X)}\right) \leq \frac{\left|\hat{G} - G(X)\right|}{1 - G(X)}.
\end{aligned} \tag{7}$$

Observe that $G(X) \in [\frac{1}{k}, 1]$ for any random variable X with support size k. Now we consider two cases. First, assume $2^b > k$. Since

$$n \geq \frac{24k^2b^2 \log \frac{2}{\delta}}{\alpha^2\epsilon^2 \min\{k, 2^b\}} = \frac{6b^2 \log \frac{2}{\delta}}{\alpha^2(\frac{\epsilon}{2})^2 \frac{1}{k}} \geq \frac{6b^2 \log \frac{2}{\delta}}{\alpha^2(\frac{\epsilon}{2})^2 \left(\left(1 - \frac{1}{2^b}\right) \frac{1}{k} + \frac{1}{2^b}\right)} \geq \frac{6b^2 \log \frac{2}{\delta}}{\alpha^2(\frac{\epsilon}{2})^2 \left(\left(1 - \frac{1}{2^b}\right) (1 - G(X)) + \frac{1}{2^b}\right)}$$

we have by Lemma G.1 that

$$\left|\hat{G} - G(X)\right| \leq \frac{\epsilon}{2} \left(1 - G(X) + \frac{1}{2^b - 1}\right) \leq \epsilon(1 - G(X)),$$

where the second inequality uses the fact that $2^b > k$ implies $\frac{1}{2^b-1} \leq \frac{1}{k} \leq 1 - G(X)$. Combining with Eq.(7) we have $|\hat{C} - C(X)| \leq \frac{\left|\hat{G} - G(X)\right|}{1 - G(X)} \leq \epsilon$.

Next, assume $2^b \leq k$. We have

$$n \geq \frac{24b^2k^2 \log \frac{2}{\delta}}{\alpha^2\epsilon^2 \min\{k, 2^b\}} = \frac{6b^2 \log \frac{2}{\delta}}{\alpha^2(\frac{\epsilon}{2k})^2 2^b} \geq \frac{6b^2 \log \frac{2}{\delta}}{\alpha^2(\frac{\epsilon}{2k})^2 \left(\left(1 - \frac{1}{2^b}\right) (1 - G(X)) + \frac{1}{2^b}\right)},$$

where the second inequality uses $2^b \geq 1$ and $1 - G(X) \leq 1$. Thus by Lemma G.1

$$\left|\hat{G} - G(X)\right| \leq \frac{\epsilon}{2k} \left(1 - G(X) + \frac{1}{2^b - 1}\right) \leq \frac{\epsilon}{k}.$$

Combining with Eq. (7) we have

$$|\hat{C} - C(X)| \leq \frac{\left|\hat{G} - G(X)\right|}{1 - G(X)} \leq k \left|\hat{G} - G(X)\right| \leq \epsilon.$$