Private Protocols for U-Statistics in the Local Model and Beyond

James Bell, Aurélien Bellet, Adrià Gascón, Tejas Kulkarni

Private Protocols for U-Statistics in the Local Model and Beyond

James Bell
The Alan Turing Institute

Aurélien Bellet
Inria

Adrià Gascón
Google

Tejas Kulkarni
Aalto University

Abstract

In this paper, we study the problem of computing U-statistics of degree 2, i.e., quantities that come in the form of averages over pairs of data points, in the local model of differential privacy (LDP). The class of U-statistics covers many statistical estimates of interest, including Gini mean difference, Kendall's tau coefficient and Area under the ROC Curve (AUC), as well as empirical risk measures for machine learning problems such as ranking, clustering and metric learning. We first introduce an LDP protocol based on quantizing the data into bins and applying randomized response, which guarantees an ϵ-LDP estimate with a Mean Squared Error (MSE) of $O(1/\sqrt{n}\epsilon)$ under regularity assumptions on the U-statistic or the data distribution. We then propose a specialized protocol for AUC based on a novel use of hierarchical histograms that achieves MSE of $O(\alpha^3/n\epsilon^2)$ for arbitrary data distribution. We also show that 2-party secure computation allows to design a protocol with MSE of $O(1/n\epsilon^2)$, without any assumption on the kernel function or data distribution and with total communication linear in the number of users n. Finally, we evaluate the performance of our protocols through experiments on synthetic and real datasets.

1 INTRODUCTION

The problem of collecting aggregate statistics from a set of n users in a way that individual contributions remain private even from the data analysts has recently attracted a lot of interest. In the popular *local model* of differential privacy (LDP) (Duchi et al., 2013; Kairouz et al., 2014), users apply a local randomizer to their private input before sending it to an untrusted aggregator. In this context, most work has focused on computing quantities that are separable across individual users, such as sums and histograms (see Bassily and Smith, 2015; Wang et al., 2017; Kulkarni et al., 2019; Cormode et al., 2018; Bassily et al., 2017, and references therein).

In this paper, we study the problem of privately computing U-statistics of degree 2, which generalize sample mean statistics to *averages over pairs of data points*. Let $x_1, \dots, x_n$ be a set of n data points drawn i.i.d. from an unknown ditribution μ over a (discrete or continuous) domain $\mathcal{X}$. The U-statistic of degree 2 with kernel f, given by $U_{f,n} = \frac{2}{n(n-1)} \sum_{i<j} f(x_i, x_j)$, is an unbiased estimate of $U_f = \mathbb{E}_{x,x' \sim \mu}[f(x, x')]$ with minimum variance (Hoeffding, 1948). The class of U-statistics covers many statistical estimates of interest, including sample variance, Gini mean difference, Kendall's tau coefficient, Wilcoxon Mann-Whitney hypothesis test and Area under the ROC Curve (AUC) (Lee, 1990; Mann and Whitney, 1947; Faivishevsky and Goldberger, 2008). They are also commonly used as empirical risk measures for machine learning problems such as ranking, clustering and metric learning (Kar et al., 2013; Clémençon et al., 2016).

Interestingly, private estimation of U-statistics in the LDP model for arbitrary kernel functions f and data distributions μ cannot be straightforwardly addressed by resorting to standard local randomizers such as the Laplace mechanism or randomized response. Indeed, one cannot apply the local randomizer to the terms of the sum based on the sensitivity of f (as each term is shared across two users), and perturbing the inputs themselves can lead to large errors when passed through the (potentially discontinuous) function f.

In this work, we design and analyze several protocols for computing U-statistics with privacy and utility guarantees. More precisely:

1. We introduce a generic LDP protocol based on quantizing the data into k bins and applying k-ary randomized response. We show that under an assumption on either the kernel function f or the

Proceedings of the 23$^{\text{rd}}$International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108. Copyright 2020 by the author(s).

data distribution μ, the aggregator can construct an ϵ-LDP estimate of $U_{f,n}$ with a Mean Squared Error (MSE) of $O(1/\sqrt{n}\epsilon)$.

2. For the case of the AUC on a domain of size 2^α, whose kernel does not satisfy the regularity assumption required by our previous protocol, we design a specialized protocol based on hierarchical histograms that achieves MSE $O(\alpha^2 \log(1/\delta)/n\epsilon^2)$ under (ϵ, δ)-LDP and $O(\alpha^3/n\epsilon^2)$ under ϵ-LDP, for arbitrary data distribution.
3. Under a slight relaxation of the local model in which we allow pairs of users i and j to compute a randomized version of $f(x_i, x_j)$ with 2-party secure computation, we show that we can design a protocol with MSE of $O(1/n\epsilon^2)$, without any assumption on the kernel function or data distribution and with constant communication for each of the n users.
4. To evaluate the practical performance of the proposed protocols, we present some experiments on synthetic and real datasets for the task of computing AUC and Kendall's tau coefficient.

The paper is organized as follows. Section 2 gives some background on U-statistics and local differential privacy. In Section 3 we present a generic LDP protocol based on randomizing quantized inputs. Section 4 introduces a specialized LDP protocol for computing the Area under the ROC Curve (AUC). In Section 5, we introduce a generic protocol which operates in a slightly relaxed version of the LDP model where users can run secure 2-party computation. We present some numerical experiments in Section 6, and conclude with a discussion of our results and future work in Section 7.

2 BACKGROUND

In this section, we introduce some background on U-statistics and local differential privacy.

2.1 U-Statistics

2.1.1 Definition and Properties

Let μ be an (unknown) distribution over an input space $\mathcal{X}$ and $f : \mathcal{X}^2 \to \mathbb{R}$ be a pairwise function (assumed to be symmetric for simplicity) referred to as the *kernel.* Given a sample $\mathcal{S} = \{x_i\}_{i=1}^n$ of n observations drawn from μ, we are interested in estimating the following population quantity:

$$U_f = \mathbb{E}_{X_1, X_2 \sim \mu}[f(X_1, X_2)]. \tag{1}$$

Definition 1 (Hoeffding, 1948)**.** *The U-statistic of degree 2 with kernel f is given by*

$$U_{f,n} = \tfrac{2}{n(n-1)} \textstyle\sum_{i<j} f(x_i, x_j). \tag{2}$$

$U_{f,n}$ is an unbiased estimate of U_f. Denoting by $\zeta_1 = \mathrm{Var}(f(x_1, X_2) \mid x_1))$ and $\zeta_2 = \mathrm{Var}(f(X_1, X_2)$, its variance is given by (Hoeffding, 1948; Lee, 1990):

$$\mathrm{Var}(U_{f,n}) = \tfrac{2}{n(n-1)}(2(n-2)\zeta_1 + \zeta_2). \tag{3}$$

The above variance is of $O(1/n)$ and is optimal among all unbiased estimators of U_f that can be computed from $\mathcal{S}$. This incurs a complex dependence structure, as each data point appears in $n-1$ pairs. The statistical behavior of U-statistics can be investigated using linearization techniques (Hoeffding, 1948) and decoupling methods (de la Pena and Giné, 1999), which provide tools to reduce their analysis to that of standard i.i.d. averages. One may refer to (Lee, 1990) for asymptotic theory of U-statistics, to (Van Der Vaart, 2000) (Chapter 12 therein) and (de la Pena and Giné, 1999) for nonasymptotic results, and to (Clémençon et al., 2008, 2016) for an account of U-statistics in the context of machine learning and empirical risk minimization.

2.1.2 Motivating Examples

U-statistics are commonly used as point estimators of various global properties of distributions, as well as in statistical hypothesis testing (Lee, 1990; Mann and Whitney, 1947; Faivishevsky and Goldberger, 2008). They also come up as empirical risk measures in machine learning problems with pairwise loss functions such as bipartite ranking, metric learning and clustering. Below, we give some concrete examples of U-statistics of broad interest to motivate our private protocols.

Gini mean difference. This is a classic measure of dispersion which is often seen as more informative than the variance for some distributions (Yitzhaki, 2003). Letting $\mathcal{X} \subset \mathbb{R}$, it is defined as

$$G = \tfrac{2}{n(n-1)} \textstyle\sum_{i<j} |x_i - x_j|, \tag{4}$$

which is a U-statistic of degree 2 with kernel $f(x_i, x_j) = |x_i - x_j|$. Gini coefficient, the most commonly used measure of inequality, is obtained by multiplying G by $(n-1)/2 \sum_{i=1}^n x_i$.

Remark 1. *The variance of a sample, obtained by replacing the absolute difference by the squared difference in* (4)*, is also a U-statistic. However we note that computing the variance can be achieved by computing two sums of locally computable variables (x_i and x_i^2), which can be done with existing LDP protocols.*

Rényi-2 entropy. Also known as collision entropy, this provides a measure of entropy between Shannon's entropy and min entropy which is used in many applications involving discrete distributions (see Acharya et al., 2015, and references therein). It is given by

$$H_2 = -\ln\left(\tfrac{2}{n(n-1)}\sum_{i<j}\mathbb{I}[x_i = x_j]\right). \qquad (5)$$

The expression inside the log is a U-statistic of degree 2 with kernel $f(x_i, x_j) = \mathbb{I}[x_i = x_j]$.

Kendall's tau coefficient. This statistic measures the ordinal association between two variables and is often used as a test statistic to answer questions such as "does a higher salary make one happier?". In learning to rank applications, it is used to evaluate the extent to which a predicted ranking correlates with the (human-generated) gold standard (see e.g., Joachims, 2002; Lapata, 2006). Formally, assuming continuous variables for simplicity, let $\mathcal{X} \subset \mathbb{R}^2$ and $\mathcal{S} = \{x_i = (y_i, z_i)\}_{i=1}^n$. For any $i < j$, the pairs $x_i = (y_i, z_i)$ and $x_j = (y_j, z_j)$ are said be *concordant* if $(y_i > y_j) \wedge (z_i > z_j)$ or $(y_i < y_j) \wedge (z_i < z_j)$, and *discordant* otherwise. Let C and D be the number of concordant and discordant pairs in $\mathcal{S}$. Kendall rank correlation coefficient is defined as:

$$\tau = \frac{C - D}{C + D} = \frac{1}{\binom{n}{2}}\sum_{i<j}\operatorname{sign}(y_i - y_j)\operatorname{sign}(z_i - z_j), \quad (6)$$

which is a U-statistic of degree 2 with kernel $f(x_i, x_j) = \operatorname{sign}(y_i - y_j)\operatorname{sign}(z_i - z_j)$.[1]

Area under the ROC curve (AUC). In binary classification with class imbalance, the Receiver Operating Characteristic (ROC) gives the true positive rate with respect to the false positive rate of a predictor at each possible decision threshold. The AUC is a popular summary of the ROC curve which gives a single, threshold-independent measure of the classifier goodness: it corresponds to the probability that the predictor assigns a higher score to a randomly chosen positive point than to a randomly chosen negative one. AUCs are widely used as performance metrics in machine learning (Bradley, 1997; Herschtal and Raskutti, 2004), and have also been recently studied as fairness measures (Kallus and Zhou, 2019; Vogel et al., 2020). Formally, let $\mathcal{X} \subset \mathbb{R} \times \{-1, 1\}$ and $\mathcal{S} = \{x_i = (s_i, y_i)\}_{i=1}^n$ where for each data point i, $s_i \in \mathbb{R}$ is the score assigned to point i and $y_i \in \{-1, 1\}$ is its label. For convenience, let $\mathcal{S}^+ = \{s_i : y_i = 1\}$ and $\mathcal{S}^- = \{s_i : y_i = -1\}$ and let $n^+ = |\mathcal{S}^+|$ and $n^- = |\mathcal{S}^-|$. The AUC is given by

$$AUC = \tfrac{1}{n^+ n^-}\sum_{s_i \in \mathcal{S}^+}\sum_{s_j \in \mathcal{S}^-}\mathbb{I}[s_i > s_j], \qquad (7)$$

where $\mathbb{I}[\sigma]$ is an indicator variable outputting 1 if the predicate σ is true and 0 otherwise. Up to a $\binom{n}{2}/n^+n^-$ factor, it is easy to see that AUC is a U-statistic of degree 2 with kernel $f(x_i, x_j) = \mathbb{I}[s_i > s_j \wedge y_i > y_j] + \mathbb{I}[s_i < s_j \wedge y_i < y_j]$.

Machine learning with pairwise losses. Many machine learning problems involve loss functions that operate on pairs of points (Kar et al., 2013; Clémençon et al., 2016). This is the case for instance in metric learning (Bellet et al., 2015), bipartite ranking (Clémençon et al., 2008) and clustering (Clémençon, 2014). Empirical risk minimization problems have therefore the following generic form:

$$\min_{\theta \in \Theta} \tfrac{2}{n(n-1)}\sum_{i<j}\ell_\theta(x_i, x_j), \qquad (8)$$

where $\theta \in \Theta$ are model parameters. The objective function in (8), as well as its gradient, are U-statistics of degree 2 with kernels ℓ_θ and $\nabla_\theta \ell_\theta$ respectively.

2.2 Local Differential Privacy

The classic *centralized* model of differential privacy assumed the presence of a trusted aggregator which processes the private information of individuals and releases a perturbed version of the result. The *local* model instead captures the setting where individuals do not trust the aggregator and randomize their input locally before sharing it. This model has received wide industrial adoption (Erlingsson et al., 2014; Fanti et al., 2016; Differential Privacy Team, Apple, 2017; Ding et al., 2017).

Definition 2 (Duchi et al., 2013). *A local randomizer $\mathcal{R}$ is (ϵ, δ)-locally differentially private (LDP) if for all $x, x' \in \mathcal{X}$ and all possible output O in the range of $\mathcal{R}$:*

$$Pr[\mathcal{R}(x) = O] \le e^\epsilon Pr[\mathcal{R}(x') = O] + \delta.$$

The special case $\delta = 0$ is called pure ϵ-LDP.

Most work in LDP aims to compute quantities that are separable across individual inputs, such as sums and histograms (see Bassily and Smith, 2015; Wang et al., 2017; Kulkarni et al., 2019; Cormode et al., 2018; Bassily et al., 2017, and references therein). In contrast, our goal is to design LDP protocols for computing U-statistics, where each term involves a pair of inputs.

3 GENERIC LDP PROTOCOL FROM QUANTIZATION

Discrete inputs. We first consider the case of discrete inputs taking one of k values. The possible values of the kernel function can be written as a matrix $A \in \mathbb{R}^{k \times k}$ where $A_{ij} = f(i, j)$. In this case, we can set the local randomizer $\mathcal{R}$ to be k-ary randomized response to generate a perturbed version $\mathcal{R}(x_i)$ of each input x_i. Let e_i denote the vector of length k with a one in the i-th position and 0 elsewhere. For each perturbed input in one-hot encoding form $e_{\mathcal{R}(x_i)}$ we can deduce an unbiased estimate of e_{x_i}. As the discrete U-statistic is

[1] One can easily modify the kernel to account for ties.

Algorithm 1: LDP algorithm based on quantization and private histograms

```
**Public Parameters:** Privacy budget $\epsilon$, quantization scheme $\pi$, number of bins $k$.
**Input:** $(x_i \in \mathcal{X})_{i \in [n]}$
**Output:** Estimate $\widehat{U}_{f,n}$ of $U_f$
for each user $i \in [n]$ do
    Form quantized input $\pi(x_i) \in [k]$
    For $\beta = k/(k + e^\epsilon - 1)$, generate $\tilde{x}_i \in [k]$ s.t.
        $P(\tilde{x}_i = i) = \begin{cases} 1-\beta & \text{for } i = \pi(x_i), \\ \beta/k & \text{for } i \neq \pi(x_i), \end{cases} \quad (9)$
    Send $\tilde{x}_i$ to the aggregator
end
Return $\widehat{U}_{f,n}$ computed from $\tilde{x}_1, \ldots, \tilde{x}_n$ and $\beta$
```

a linear function of each of these vectors, computing it on these unbiased estimates gives an unbiased estimate $\widehat{U}_{f,n}$ which can be written as:

$$\widehat{U}_{f,n} = \frac{1}{\binom{n}{2}} \sum_{1 \leq i < j \leq n} \widehat{f}_A(\mathcal{R}(x_i), \mathcal{R}(x_j)),$$

and is itself a U-statistic with kernel $\widehat{f}_A$ given by

$$\widehat{f}_A(\mathcal{R}(x_1), \mathcal{R}(x_2)) = (1-\beta)^{-2}(e_{\mathcal{R}(x_1)} - b)^T A(e_{\mathcal{R}(x_2)} - b),$$

where $1-\beta$ is the probability of returning the true input in k-ary randomized response (see Eq. 9) and b is the vector of length k with every entry β/k. Details and analysis of this process, leveraging Hoeffding's decomposition of U-statistics (Hoeffding, 1948; Lee, 1990), can be found in Section A.1 of the supplementary material. The resulting bounds on the variance of $\widehat{U}_{f,n}$ are summarized in the following theorem.

Theorem 1. *If $f(x, x') \in [0,1]$ for all x, x', then*

$$\mathrm{Var}(\widehat{U}_{f,n}) \leq \frac{1}{n(1-\beta)^2} + \frac{(1+\beta)^2}{2n(n-1)(1-\beta)^4}.$$

In order to achieve ϵ-LDP with a fixed k this becomes,

$$\mathrm{Var}(\widehat{U}_{f,n}) \approx \frac{(1+k/\epsilon)^2}{n} + \frac{(1+k/\epsilon)^4}{2n^2} \approx \frac{k^2}{n\epsilon^2},$$

Continuous inputs. For U-statistics on discrete domains, e.g. Renyi-2 entropy, the above strategy can be applied directly. Possibly more importantly however, this then leads to a natural protocol for the continuous case. In this protocol (see Algorithm 1), the local randomizer proceeds by quantizing the input into k bins (for instance using simple or randomized rounding) before applying the previous procedure. There are two sources of error in this protocol. The first one is due to the randomization needed to satisfy LDP in the quantized domain as bounded in Theorem 1. The second source of error is due to quantization. In order to control this error in a nontrivial way, we rely on an assumption on the kernel function (namely, that it is Lipschitz) or the data distribution (namely, that it has Lipschitz density). Under these assumptions and using an appropriate variant of the kernel function on the quantized domain, we show that we can bound the error with respect to the original domain by a term in $O(1/k^2)$ (see Section A.2 of the supplementary material). This leads to the following result.

Theorem 2. *For simplicity, assume bounded domain $\mathcal{X} = [0,1]$ and kernel values $f(x,y) \in [0,1]$ for all $x, y \in \mathcal{X}$. Let π correspond to simple rounding, $\epsilon > 0$, $k \geq 1$ and $\beta = k/(k + e^\epsilon - 1)$. Then Algorithm 1 satisfies ϵ-LDP. Furthermore:*

- *If f is L_f-Lipschitz in each of its arguments, then $\mathtt{MSE}(\widehat{U}_{f,n})$ is less than or equal to*
$$\frac{1}{n(1-\beta)^2} + \frac{(1+\beta)^2}{2n(n-1)(1-\beta)^4} + \frac{L_f^2}{2k^2}.$$
- *If $d\mu/d\lambda$ is L_μ-Lipschitz w.r.t. some measure λ, then $\mathtt{MSE}(\widehat{U}_{f,n})$ is less than or equal to*
$$\frac{1}{n(1-\beta)^2} + \frac{(1+\beta)^2}{2n(n-1)(1-\beta)^4} + \frac{4L_\mu^2}{k^2} + \frac{4L_\mu^4}{k^4}.$$

Remark 2. *The use of simple rounding is not optimal in many situations. In the case of sum, and possibly of the Gini coefficient, it would be more accurate if randomized rounding were used instead of simple rounding. We leave this investigation for later work.*

Setting k so as to balance the quantization and estimation errors leads to the following corollary.

Corollary 1. *Under the conditions of Theorem 2, for $\epsilon \leq 1$ and large enough n, taking $k = n^{1/4}\sqrt{L\epsilon}$ leads to $\mathtt{MSE}(\widehat{U}_{f,n}) = O(L/\sqrt{n}\epsilon)$, where L corresponds to L_f or L_μ depending on the assumption.*

This result gives concrete error bounds for U-statistics whose kernel is Lipschitz, for arbitrary data distributions. One important example is the Gini mean difference, whose corresponding kernel $f(x_i, x_j) = |x_i - x_j|$ is 1-Lipschitz. On the other hand, for U-statistics with non-Lipschitz kernels, the data distribution must be sufficiently smooth (if not, it is easy to construct cases that make the algorithm fail).

4 LOCALLY PRIVATE AUC

In this section, we describe an algorithm for computing AUC (7), whose kernel is discontinuous and therefore

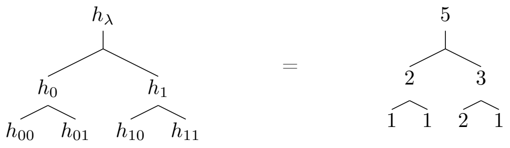

Figure 1: Hierarchical histogram h for multiset $\{0, 1, 2, 2, 3\}$ over the domain $\{0, 1, 2, 3\}$.

non-Lipschitz. We assume $\mathcal{X}$ to be an ordered domain of size d, that is with each datum in $[0..d-1]$. Note that all data is in practice discrete when represented in finite precision, so this is general. For simplicity of presentation we will assume that (i) $d = 2^\alpha$ for some integer α, and (ii) that the classes of the data, the y_i, are public.

Our solution for computing AUC in the local model relies on a hierarchical histogram construction that has been considered in previous works for private collection of high-dimensional data (Chan et al., 2012), heavy hitters (Bassily et al., 2017), and range queries (Kulkarni et al., 2019). A hierarchical histogram is essentially a tree data structure on top of a histogram where each internal node is labelled with the sum of the values in the interval covered by it (see Figure 1). That allows to answer any range query about u by checking the value associated with $O(\log |u|)$ nodes in the tree. We first define an exact version of such hierarchical histograms and explain how to compute AUC from one.

Notation on trees. We represent a binary tree h of depth α with integer node labels as a total mapping from a prefix-closed set of binary strings of length at most α to the integers. We refer to the i-th node in level l of the tree by the binary representation of i padded to length l from the left with zeros. With this notation, h_λ is the label of the root node, as we use λ to denote the empty string, h_0 (resp. h_1) is the integer label of the left (resp. right) child of the root of h, and in general h_p is the label of the node at path p from the root, i.e. the label of the node reached by following left or right children from the root according to the value of p (0 indicates left and 1 indicates right). Let b_i be the i-th node in the bottom level. For two binary strings $p, p' \in \{0,1\}^*$ we denote the prefix relation by $p' \preceq p$, and their concatenation as $p \cdot p'$.

Definition 3. *Let $S = \{s_1, \ldots, s_n\}$ be a multiset, with $s_i \in [0..d-1]$. A hierarchical histogram of S is a total mapping $h : \{0,1\}^{\leq \log(d)} \to \mathbb{Z}$ defined as $h(b) = |\{s \in S \mid \exists b' \in \{0,1\}^* : b \cdot b' = b_s\}|$. For simplicity, we denote $h(b)$ by h_b.*

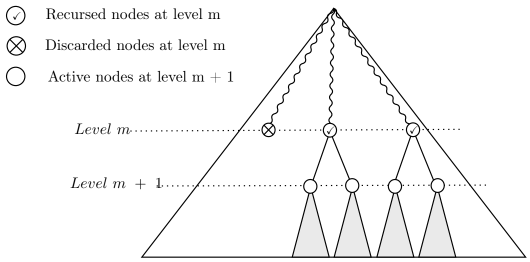
Recursed nodes at level m
Discarded nodes at level m
Active nodes at level m + 1
Level m
Level m + 1

Figure 2: Our algorithm can be seen as a breath-first traversal of a tree, where at each level some nodes are selected for their subtrees to be explored further.

Algorithm. We use hierarchical histograms to compute AUC as follows. Let S^+ and S^- be the samples of the positive and negative classes from which we want to estimate AUC. Let h^+ and h^- be hierarchical histograms for S^+ and S^-. Note that $h^+_\lambda = n^+$ and $h^-_\lambda = n^-$. We can now define the unnormalized AUC, denoted UAUC, over hierarchical histograms recursively by letting $\texttt{UAUC}(h^+, h^-, p)$ be 0, if p is a leaf, and otherwise setting:

$$\texttt{UAUC}(h^+, h^-, p) = h^+_{p\cdot 1} h^-_{p\cdot 0} + \sum_{i \in \{0,1\}} \texttt{UAUC}(h^+, h^-, p\cdot i) \ .$$

Thus we have $\texttt{AUC}(S^+, S^-) = \texttt{AUC}(h^+, h^-, \lambda) = \frac{1}{n^+ n^-} \texttt{UAUC}(h^+, h^-, \lambda)$.

The above definition naturally leads to an algorithm that proceeds by traversing the trees h^+, h^- top-down from the root λ, accumulating the products of counts from h^+, h^- at nodes that correspond to entries in h^+ that are bigger than entries in h^-. We now define a differentially private analogue. Later we will describe an efficient frequency oracle which can be used to compute an LDP estimate $\hat{h}$ of a hierarchical histogram h of n values in a domain of size 2^α. This will provide the following necessary properties (i) $\hat{h}$ is *unbiased*, (ii) $\mathrm{Var}(\hat{h}) \leq v$, with v defined as $Cn\alpha$ for some small constant C (iii) the $\hat{h}_p$ are pairwise independent and (iv) Each level of $\hat{h}$ is independent of the other levels. Our private algorithm for computing an estimate of UAUC is then defined in terms of parameters n^+ and n^-, v^+ and v^- (bounding the variance of $\hat{h}^+$ and $\hat{h}^-$ respectively), and $a > 1$ is a small number depending on n^+, n^-, α and C.

For a symbol $\aleph$ we write $\aleph^\pm$ to simultaneously refer to $\aleph^+$ and $\aleph^-$. Let $\tilde{h}^\pm_p = \max(\hat{h}^\pm_p, \sqrt{av^\pm}/2)$, i.e. $\tilde{h}^+_p = \max(\hat{h}^+_p, \sqrt{av_+}/2)$ and $\tilde{h}^-_p = \max(\hat{h}^-_p, \sqrt{av_-}/2)$, and let $\tau = a\sqrt{v^- v^+}$. Our private estimate is defined as follows. If p is a leaf then $\widehat{\texttt{UAUC}}(\hat{h}^+, \hat{h}^-, p)$ is 0, else if $\tilde{h}^+_p \tilde{h}^-_p < \tau$ then it is given by

$$\tfrac{1}{2} \textstyle\sum_{i \in \{0,1\}} \hat{h}^+_{p\cdot i} \sum_{i \in \{0,1\}} \hat{h}^-_{p\cdot i} \ .$$

Otherwise, it is given recursively by

$$\hat{h}^+_{p\cdot 1} \hat{h}^-_{p\cdot 0} + \textstyle\sum_{i \in \{0,1\}} \widehat{\texttt{UAUC}}(\hat{h}^+, \hat{h}^-, p \cdot i) \ . \tag{10}$$

As before, this definition leads to an algorithm. Note that the only difference with its non-private analogue is that this procedure does not recurse into subtrees whose contribution to the UAUC is upper bounded sufficiently tightly. More concretely, the server starts by querying $\hat{h}^+, \hat{h}^-$ at the root, namely with $p = \lambda$. If p is a leaf then we return 0 as the AUC. Otherwise, the algorithm checks whether $\tilde{h}_p^+\tilde{h}_p^- < \tau$. If so, then the algorithm concludes that there is not much to gain in exploring the subtrees rooted at $p{\cdot}0$ and $p{\cdot}1$, and returns $\frac{1}{2}\sum_{i\in 0,1}\hat{h}^+_{p\cdot i}\sum_{i\in 0,1}\hat{h}^-_{p\cdot i}$ as an estimate of $\frac{1}{2}h_p^+h_p^-$. This estimate might seem equivalent to $\frac{1}{2}\hat{h}_p^+\hat{h}_p^-$, but takes the previous form for a technical reason that is made clear in the proof. In this case we call p a *discarded* node. On the other hand, if $\tilde{h}_p^+\tilde{h}_p^- \geq \tau$, the algorithm proceeds as its non-private analogue, accumulating the contribution to the `UAUC` from the direct subtrees of p and recursing into nodes $p \cdot 0$ and $p \cdot 1$. In this case we refer to p as a *recursed* node. Thus every node $p \in \{0,1\}^{\leq\alpha}$ will be either recursed, a leaf or there will be a discarded node p' such that $p' \preceq p$. This is depicted in Figure 2.

Analysis. Note that our algorithm has two sources of error: (i) the one incurred by discarding nodes and (ii) the error in estimating the contribution to the UAUC of the recursed nodes. The threshold τ is carefully chosen to balance these two errors.

Let R^m be the set of nodes recursed on at level m. Our accuracy proof starts by bounding the expected value of $|R^m|$ (see Lemma 4 in Section B.1 of the supplementary) by a quantity B that is independent of m. We now describe a central argument to our accuracy proof, stated in the next theorem. Let E^R_m be the contribution to the error by nodes in R^m. Then, the total contribution to the error by recursed nodes is $E^R = \sum_{m\in[\alpha]} E^R_m$. A useful identity is $\mathbb{E}({E^R}^2) = \sum_{m\in[\alpha]}\mathbb{E}({E^R_m}^2)$, as we can bound $\mathbb{E}({E^R_m}^2)$, for any m, in terms of B (see detailed proof in the supplementary). Note that this identity follows from $\mathbb{E}(E^R_m E^R_{m'}) = 0$, with $m' > m$. The latter would hold if errors E^R_m and $E^R_{m'}$ were independent, since our frequency oracle is unbiased. However, errors at a given level are not independent of previous levels. However $\mathbb{E}(E^R_m E^R_{m'}) = 0$ because the conditional expectation of $E^R_{m'}$ with respect to the answers of the frequency oracles up to level m' is 0 i.e. $E^R_1, \dots, E^R_{m'}$ is a martingale difference sequence. The idea of conditioning on previous levels is used several times in our proofs, also to bound the error due to discarded nodes.

Next, we state our accuracy result, which is proven in detail in Section B.1 of the supplementary. Our proof tracks constants: this is important for practical purposes, and we show empirically in Section D.1 that our bound is in fact quite tight.

Theorem 3. *If $\alpha \leq \sqrt{n}$ and the following holds:*

1. *$\mathbb{E}(\hat{h}_p^\pm - h_p^\pm) = 0$ i.e. frequency estimates are unbiased.*
2. *$\mathbb{E}((\hat{h}_p^\pm - h_p^\pm)^2) \leq v^\pm$ i.e. `MSE` of frequency estimator is bounded by $v^\pm = Cn^\pm\alpha$.*
3. *For distinct $p, p' \in \{0,1\}^{\leq\alpha}$ with $|p| = |p'|$, $\hat{h}_p^\pm$ and $\hat{h}_{p'}^\pm$ are independent i.e. the frequency estimates are pairwise independent.*
4. *For all $m \leq \log(d)$, the lists $(\hat{h}_p^\pm)_{p\in\{0,1\}^{\leq m}}$ and $(\hat{h}_p^\pm)_{p\in\{0,1\}^{>m}}$ are independent of each other.*

Then, `MSE`$(\widehat{\mathtt{UAUC}})$ is given by

$$Cn^-n^+\alpha^2\Big(2n + (4a+1)\min(n^-, n^+) + \frac{21\sqrt{2nC\alpha}}{\sqrt{a}-1}\Big)$$

Instantiating $\hat{h}$. So far, Theorem 3 does not yield a complete algorithm as it does not specify an algorithm for computing estimates $\hat{h}$ of a hierarchical histogram that satisfy the conditions of Theorem 3. In Section B.2 of the supplementary, we show how to instantiate such an algorithm in a communication-efficient manner by combining ideas from Bassily et al. (2017), in particular the use of the Hadamard transform, with an modified version of the protocol from Kulkarni et al. (2019). This leads to the following result.

Theorem 4. *There is a one-round non-interactive protocol for computing AUC in the local model with `MSE` bounded by $O(\alpha^2\log(1/\delta)/n\epsilon^2)$ under (ϵ,δ)-LDP and $O(\alpha^3/n\epsilon^2)$ under ϵ-LDP. Every user submits one bit, and the server does $O(nlog(d))$ computation and requires $O(log(d))$ additional reconstruction space.*

5 GENERIC PROTOCOLS FROM 2PC

So far, we have proposed a specialized LDP protocol for AUC, and a generic LDP protocol which requires some assumption on the kernel function or the data distribution to guarantee nontrivial error bounds. We conjecture that no LDP protocol can guarantee nontrivial error for arbitrary kernels and distributions, but we leave this as an open question for future work.

In this section, we slightly relax the model of LDP by allowing pairs of users i and j to compute a randomized version $\tilde{f}(x_i, x_j)$ of their kernel value $f(x_i, x_j)$ with 2-party secure computation (2PC). This gives rise to a computational differential privacy (CDP) model Mironov et al. (2009). Unsurprisingly, we show that in this model we can match the MSE of $O(\frac{\ln(1/\delta)}{n\epsilon^2})$ for computing regular (univariate) averages in the (ϵ, δ)-LDP model by using advanced composition results (Dwork et al., 2010). However, such a protocol requires $O(n^2)$ communication as all pairs of users need to compute $\tilde{f}(x_i, x_j)$ via 2PC, and does not satisfy pure ϵ-DP.

Proposed protocol. To address these limitations, we propose that the aggregator asks only a (random) subset of pairs of users (i, j) to submit their randomized kernel value $\tilde{f}(x_i, x_j)$. The idea is to trade-off between the error due to privacy (which increases as more pairs are used, due to budget splitting) and the *subsampling error* (for not averaging over all pairs). Given a positive integer P (which should be thought of as a small constant independent of n) and assuming n to be even for simplicity, we propose the following protocol:

1. *Subsampling:* The aggregator samples P independent permutations $\sigma_1, \dots, \sigma_P \in \mathfrak{S}_n$ of the set of users $\{1, \dots, n\}$. This defines a set of $Pn/2$ pairs $\mathcal{P} = (\sigma_p(2i-1, 2i))_{p\in[P], 1\leq i\leq N/2}$.
2. *Perturbation:* For each pair of users $(i, j) \in \mathcal{P}$, users compute $\tilde{f}(x_i, x_j)$ via 2PC and sends it to the aggregator.
3. *Aggregation:* The aggregator computes an estimate of U_f as a function of $\{\tilde{f}(x_i, x_j)\}_{(i,j)\in\mathcal{P}}$.

Analysis. We have the following result for the Laplace mechanism applied to real-valued kernel functions (the extension to randomized response for discrete-valued kernels is straightforward). The proof relies on an exact characterization of the subsampling error by leveraging results on the variance of incomplete U-statistics (Blom, 1976), see Section C.1 of the supplementary for details.

Theorem 5 (2PC subsampling protocol with Laplace mechanism)**.** *Let $\epsilon > 0$, $P \geq 1$ and assume that the kernel f has values in $[0, 1]$. Consider our subsampling protocol above with $\tilde{f}(x_i, x_j) = f(x_i, x_j) + \eta_{ij}$ where $\eta_{ij} \sim Lap(P/\epsilon)$, and the estimate computed as $\widehat{U}_{f,n} = \frac{2}{Pn}\sum_{(i,j)\in\mathcal{P}} \tilde{f}(x_i, x_j)$. Then the protocol satisfies ϵ-CDP, has a total communication cost of $O(Pn)$ and*

$$\mathit{MSE}(\widehat{U}_{f,n}) = \frac{2}{Pn}\Big(2(P-1)\big(1 - \frac{1}{n-1}\big)\zeta_1 + \big(1 + \frac{P-1}{n-1}\big)\zeta_2\Big) + \frac{2P}{n\epsilon^2},$$

where ζ_1 and ζ_2 are defined as in (3)*.*

The MSE in Theorem 5 is of $O(\frac{1}{Pn} + \frac{P}{n\epsilon^2})$. Remarkably, this shows that the $O(1/n)$ variance of the estimate that uses all pairs is preserved when subsampling only $O(n)$ pairs. This is made possible by the strong dependence structure in the $O(n^2)$ terms of the original U-statistic. As expected, P rules a trade-off between the errors due to subsampling and to privacy: the larger P, the smaller the former but the larger the latter (as each user must split its budget across P pairs). The optimal value of P depends on the kernel function and the data distribution (through ζ_1 and ζ_2) on the one hand, and the privacy budget ϵ on the other hand. This trade-off, along with the optimality of the proposed subsampling schemes, are discussed in more details Section C.1 of the supplementary material. In practice and as illustrated in our experiments, P can be set to a small constant.

Implementing 2PC. Securely computing the randomized kernel value $\tilde{f}(x_i, x_j)$ can be done efficiently for many kernel functions and local randomizers of interest, as the number of parties involved is limited to 2. We assume semi-honest parties (see Goldreich, 2004, for a definition of this threat model). A suitable 2PC technique in this application are garbled circuits (Yao, 1986; Lindell and Pinkas, 2009; Evans et al., 2018), which are well-suited to compute Boolean comparisons as required in several of the kernels mentioned in Section 2.1.2. The circuits for computing the kernels can then be extended with output perturbation following ideas from Dwork et al. (2006) and Champion et al. (2019). We refer to Section C.2 of the supplement for details on design and complexity.

Remark 3 (Beyond 2PC)**.** *One could further relax the model to allow multi-party secure computation with more than two parties, e.g. by extending the garbled circuit computing the kernel with secure aggregation over the Pn pairs before performing output perturbation. This would recover the utility of centralized DP at the cost of much more computation and quadratic communication, which is not practical, as well as robustness. More interesting trade-offs may be achieved by securely aggregating small subsets of pairs. We leave the careful analysis of such extensions to future work.*

6 EXPERIMENTS

AUC. We use the Diabetes dataset (Strack et al., 2014) for the binary classification task of determining whether a patient will be readmitted in the next 30 days after being discharged. We train a logistic regression model s which is used to score data points in $[0, 1]$, and apply our protocol to privately compute AUC on the test set. Patients readmitted before 30 days form the positive class. Class sizes are shown in Figure 3. Class information is not considered sensitive, as opposed to the score $s(x)$ on private user data $s(x)$, which includes detailed medical information. Figure 3 shows the standard error achieved by our protocol for different values of the domain size d. For each value of d we run our protocol with inputs $s(\texttt{fp}(s(x_i), d))$, where $\texttt{fp}$

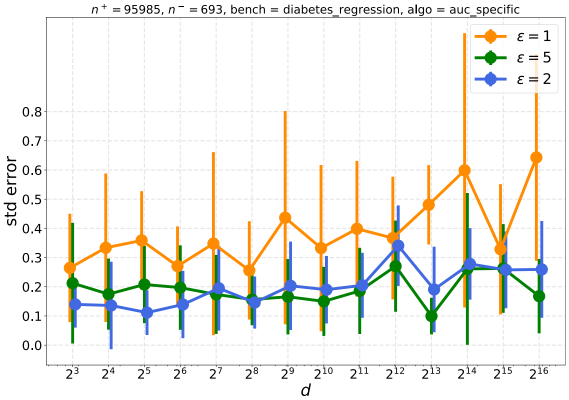

Figure 3: Mean and std. dev. (over 20 runs) of the absolute error of our AUC protocol on the scores of a logistic regression model trained on a Diabetes dataset.

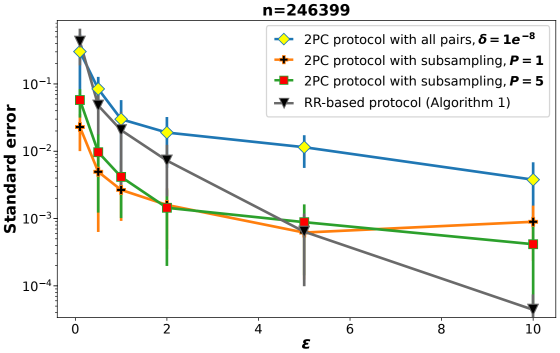

Figure 4: Mean and standard deviation (over 20 runs) of the absolute error in KTC on Tripadvisor dataset.

denotes a discretization into the domain $[0..d-1]$. The plot shows that the protocol is quite robust to the choice of d, and that increasing ϵ beyond 2 does not improve results significantly. Recall that the error of our AUC protocol depends on the size of the smallest class, which is quite small here (only 693 examples).

Kendall's tau. We use the Tripadvisor dataset (Wang, 2010). The dataset consists of discrete user ratings (from scale -1 to 5) for hotels in San Francisco over many service quality metrics such as room service, location, room cleanliness, front desk service etc. After discarding the records with missing values, we have over 246K records. Let $x_i = (y_i, z_i)$ be ratings given by user i to the room (y_i) and the cleanliness (z_i). The goal is to privately estimate the Kendall's tau coefficient (KTC) between these two variables, whose true value is 0.58. We compare the privacy-utility trade-off of our randomized response protocol (Algorithm 1 without quantization, since inputs can take only $6 \times 6 = 36$ values), our 2PC protocol based on subsampling, and the 2PC protocol that computes all pairs and relies on advanced composition (for which we set $\delta = 1e^{-8}$). The 2PC primitive to compute $\tilde{f}(x_i, x_j)$ is simulated. The results shown in Figure 4 show that the 2PC protocol with all pairs performs worst due to composition. The randomized response protocol performs slightly better, thanks to the small domain size. Finally, our 2PC protocol with subsampling achieves the lowest error by roughly an order of magnitude in high privacy regimes ($\epsilon \leq 2$) while keeping the communication cost linear in n. As predicted by our analysis, $P = 1$ is best in high privacy regimes, where the error due to privacy dominates the subsampling error. We also see that $P > 1$ can be used to reduce the overall error in low privacy regimes (for $\epsilon = 10$ one can use an even larger P to match the error of the randomized response protocol).

7 CONCLUDING REMARKS

In this paper, we tackled the problem of computing U-statistics from private user data, covering many statistical quantities of broad interest which were not addressed by previous private protocols.

Relative merits of our protocols. Our three protocols are largely complementary, insofar as each of them is well-suited to specific situations. Our first protocol (quantization followed by randomized response) can gracefully handle cases where the kernel is Lipschitz or the data is discrete in a small domain. It may also work well for non-Lipschitz kernels when quantizing data into few bins does not lose too much information (e.g., when the data distribution is smooth enough). As the latter hypothesis is difficult to assess in advance, we argue that there can exist specialized protocols that work well for non-Lipschitz kernels on continuous data or large discrete domains. Our second protocol illustrates this for the case of AUC: we leverage a hierarchical histogram structure to scale much better with the number of bins than the first protocol (see Section D for experiments comparing these two protocols). Finally, if one is willing to slightly relax the LDP model to allow pairwise communication among users, and if the kernel can be computed efficiently via 2PC, our third protocol is expected to perform best in terms of accuracy.

Extension to higher degrees. While we focused on *pairwise* U-statistics, our ideas can be extended to higher degrees. A prominent example is the Volume Under the ROC Surface, the generalization of the AUC to multi-partite ranking (Clémençon et al., 2013).

Future work. A promising direction for future work is to develop private multi-party algorithms for learning with pairwise losses (Kar et al., 2013; Clémençon et al., 2016) by combining private stochastic gradient descent for standard empirical risk minimization (Bassily et al., 2014; Shokri and Shmatikov, 2015) and our protocols to compute the gradient estimates.

Acknowledgments

J. B. was supported by The Alan Turing Institute underthe EPSRC grant EP/N510129/1, and the UK Government's Defence & Security Programme in support of the Alan Turing Institute. A. B. was supported by grants ANR-16-CE23-0016-01 and ANR-18-CE23-0018-03, by the European Union's Horizon 2020 Research and Innovation Program under Grant Agreement No. 825081 COMPRISE and by a grant from CPER Nord-Pas de Calais/FEDER DATA Advanced data science and technologies 2015-2020. A. B. thanks Jan Ramon for useful discussions. Work partly done when A. G. was at The Alan Turing Institute and Warwick University, and supported by The Alan Turing Institute under the EPSRC grant EP/N510129/1, and the UK Government's Defence & Security Programme in support of the Alan Turing Institute. T. K. thanks Graham Cormode for arranging a trip to Inria Lille. His visit was supported by Marie Curie Grant 618202.

References

Acharya, J., Orlitsky, A., Suresh, A. T., and Tyagi, H. (2015). The Complexity of Estimating Rényi Entropy. In *SODA*.

Bassily, R., Nissim, K., Stemmer, U., and Thakurta, A. G. (2017). Practical locally private heavy hitters. In *NIPS*.

Bassily, R. and Smith, A. (2015). Local, private, efficient protocols for succinct histograms. In *STOC*.

Bassily, R., Smith, A. D., and Thakurta, A. (2014). Private Empirical Risk Minimization: Efficient Algorithms and Tight Error Bounds. In *FOCS*.

Bellet, A., Habrard, A., and Sebban, M. (2015). *Metric Learning*. Morgan & Claypool Publishers.

Blom, G. (1976). Some properties of incomplete U-statistics. *Biometrika*, 63(3):573–580.

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159.

Champion, J., Shelat, A., and Ullman, J. (2019). Securely sampling biased coins with applications to differential privacy. *IACR Cryptology ePrint Archive*, 2019:823.

Chan, T. H., Shi, E., and Song, D. (2012). Privacy-preserving stream aggregation with fault tolerance. In *Financial Cryptography*.

Clémençon, S. (2014). A statistical view of clustering performance through the theory of U-processes. *Journal of Multivariate Analysis*, 124:42–56.

Clémençon, S., Bellet, A., and Colin, I. (2016). Scaling-up Empirical Risk Minimization: Optimization of Incomplete U-statistics. *Journal of Machine Learning Research*, 13:165–202.

Clémençon, S., Lugosi, G., and Vayatis, N. (2008). Ranking and empirical risk minimization of U-statistics. *The Annals of Statistics*, 36(2):844–874.

Clémençon, S., Robbiano, S., and Vayatis, N. (2013). Ranking data with ordinal labels: Optimality and pairwise aggregation. *Machine Learning*, 91(1):67–104.

Cormode, G., Kulkarni, T., and Srivastava, D. (2018). Marginal release under local differential privacy. In *SIGMOD*.

de la Pena, V. and Giné, E. (1999). *Decoupling: from Dependence to Independence*. Springer.

Differential Privacy Team, Apple (2017). Learning with privacy at scale.

Ding, B., Kulkarni, J., and Yekhanin, S. (2017). Collecting telemetry data privately. In *NIPS*.

Duchi, J. C., Jordan, M. I., and Wainwright, M. J. (2013). Local privacy and statistical minimax rates. In *FOCS*.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. (2006). Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*.

Dwork, C., Rothblum, G. N., and Vadhan, S. (2010). Boosting and Differential Privacy. In *FOCS*.

Erlingsson, U., Pihur, V., and Korolova, A. (2014). Rappor: Randomized aggregatable privacy-preserving ordinal response. In *CCS*.

Evans, D., Kolesnikov, V., and Rosulek, M. (2018). A pragmatic introduction to secure multi-party computation. *Foundations and Trends in Privacy and Security*, 2(2-3):70–246.

Faivishevsky, L. and Goldberger, J. (2008). ICA based on a Smooth Estimation of the Differential Entropy. In *NIPS*.

Fanti, G., Pihur, V., and Erlingsson, Ú. (2016). Building a RAPPOR with the unknown: Privacy-preserving learning of associations and data dictionaries. In *PoPETs*.

Goldreich, O. (2004). *The Foundations of Cryptography - Volume 2, Basic Applications*. Cambridge University Press.

Herschtal, A. and Raskutti, B. (2004). Optimising area under the ROC curve using gradient descent. In *ICML*.

Hoeffding, W. (1948). A class of statistics with asymptotically normal distribution. *Annals of Mathematics and Statistics*, 19:293–325.

Joachims, T. (2002). Optimizing search engines using clickthrough data. In *KDD*.

Kairouz, P., Oh, S., and Viswanath, P. (2014). Extremal mechanisms for local differential privacy. In *NIPS*.

Kallus, N. and Zhou, A. (2019). The Fairness of Risk Scores Beyond Classification: Bipartite Ranking and the xAUC Metric. In *NeurIPS*.

Kar, P., Sriperumbudur, B. K., Jain, P., and Karnick, H. (2013). On the Generalization Ability of Online Learning Algorithms for Pairwise Loss Functions. In *ICML*.

Kulkarni, T., Cormode, G., and Srivastava, D. (2019). Answering range queries under local differential privacy. In *SIGMOD*.

Lapata, M. (2006). Automatic Evaluation of Information Ordering: Kendall's Tau. *Computational Linguistics*, 32(4):471–484.

Lee, A. (1990). *U-statistics: Theory and practice*. Marcel Dekker, Inc., New York.

Lindell, Y. and Pinkas, B. (2009). A proof of security of yao's protocol for two-party computation. *Journal of Cryptology*, 22(2):161–188.

Mann, H. B. and Whitney, D. R. (1947). On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. *Annals of Mathematical Statistics*, 18(1):50–60.

Mironov, I., Pandey, O., Reingold, O., and Vadhan, S. P. (2009). Computational Differential Privacy. In *CRYPTO*.

Shokri, R. and Shmatikov, V. (2015). Privacy-preserving deep learning. In *CCS*.

Strack, B., DeShazo, J. P., Gennings, C., Olmo, J. L., Ventura, S., Cios, K. J., and Clore, J. N. (2014). Diabetes data. `https://archive.ics.uci.edu/ml/datasets/diabetes+130-us+hospitals+for+years+1999-2008`.

Van Der Vaart, A. W. (2000). *Asymptotic Statistics*. Cambridge University Press.

Vogel, R., Bellet, A., and Clémençon, S. (2020). Learning Fair Scoring Functions: Fairness Definitions, Algorithms and Generalization Bounds for Bipartite Ranking. *arXiv preprint arXiv:2002.08159*.

Wang, H. (2010). Trip advisor data. `http://www.preflib.org/data/combinatorial/trip/`.

Wang, T., Blocki, J., Li, N., and Jha, S. (2017). Locally differentially private protocols for frequency estimation. In *USENIX Security Symposium*.

Yao, A. C. (1986). How to generate and exchange secrets (extended abstract). In *FOCS*.

Yitzhaki, S. (2003). Gini's Mean Difference: A Superior Measure of Variability for Non-Normal Distributions. *Metron International Journal of Statistics*, 61(2):285–316.