Eureka: A General Framework for Black-box Differential Privacy Estimators

Yun Lu
University of Victoria
yunlu@uvic.ca

Malik Magdon-Ismail
Rensselaer Polytechnic Institute
magdon@cs.rpi.edu

Yu Wei
Purdue University
yuwei@purdue.edu

Vassilis Zikas
Purdue University
vzikas@purdue.edu

***Abstract*—Differential privacy (DP) is a key tool in privacy-preserving data analysis. Yet it remains challenging for non-privacy-experts to prove the DP of their algorithms. We propose a methodology for domain experts with limited data privacy background to empirically estimate the privacy of an *arbitrary* mechanism. Our Eureka moment is a new link—which we prove—between the problems of DP parameter-estimation and Bayes optimal classifiers in ML, which we believe can be of independent interest. Our estimator uses this link to achieve two desirable properties: (1) *black-box*, i.e., it does not require knowledge of the underlying mechanism, and (2) it has a theoretically-proven accuracy, depending on the underlying classifier used, allowing plug-and-play use of different classifiers.**

More concretely, motivated by the impossibility of the above task for unrestricted input domains (which we prove), we introduce a natural, application-inspired relaxation of DP which we term *relative DP*. Intuitively, relative DP defines a mechanism's privacy relative to an input set $\mathcal{T}$, circumventing the above impossibility when $\mathcal{T}$ is finite. Importantly, it preserves the key intuitive privacy guarantee of DP while enjoying a number of desirable DP properties—scalability, composition, and robustness to post-processing. We then devise a black-box poly-time (ε, δ)-relative DP estimator for *any* poly-size $\mathcal{T}$—the first privacy estimator to support mechanisms with large *output* spaces while having tight accuracy bounds. As a result of independent interest, we generalize our theory to develop the *first Distributional Differential Privacy* (DDP) estimator.

We benchmark our estimator in a proof-of-concept implementation. First, using kNN as the classifier we show that our method (1) produces a tight, analytically computed (ε, δ)-DP trade-off of low-dimensional Laplace and Gaussian mechanisms—the first to do so, (2) accurately estimates the privacy spectrum of DDP mechanisms, and (3) can verify a DP mechanism's implementations, e.g., Sparse Vector Technique, Noisy Histogram, and Noisy max. Our implementation and experiments demonstrate the potential of our framework, and highlight its computational bottlenecks in estimating DP, e.g., in terms of the size of δ and the data dimensionality. Our second, neural-network-based instantiation makes a first step in showing that our method can be extended to mechanisms with high-dimensional outputs.

1. Introduction

As big-data e.g., machine learning (ML) algorithms become more sophisticated and ubiquitous, the need to protect sensitive data becomes ever more prominent. Differential privacy (DP) is a broadly accepted privacy notion. Yet, the need to prove DP for these often complex algorithms limits the accessibility of DP to application domain experts who are not trained in security.

Informally, a mechanism $\mathcal{M}$ is (ε, δ)-DP if for any pair of *neighboring* databases D, D', the output distributions of $\mathcal{M}(D)$ and $\mathcal{M}(D')$ are (ε, δ)-close. Parameters ε and δ quantify the DP of $\mathcal{M}$, smaller is better. Intuitively, ε quantifies the privacy of each individual record, and δ is the probability that all privacy guarantees are given up. The desired δ is as small as possible, but in most applications—and for most mechanisms—there is an inherent trade-off between ε and δ–typically, tiny δ comes at the cost of large ε. Hence, knowing just a single pair of privacy parameters (ε, δ) for a mechanism may be insufficient to understand its privacy guarantees. It does not answer, for example, the question "What happens to δ (resp. ε) if I claim a smaller ε (resp. δ) for the same mechanism?".

Charting the above trade-off gives an important insight in deciding whether or not a mechanism is a good fit for some given application. Consider for example the use of DP for privacy-preserving release of surveys from the US census bureau. According to [1], a DP budget of $\varepsilon = 17.14$ was used to ensure that $\delta = 10^{-10}$. The above price (in terms of ε and/or utility) paid for maintaining such tiny δs is high and offers debatable DP guarantees. Thus, one may ask: "Is this necessary, and what are reasonable alternative (ε, δ) pairs"? Our work is motivated by our thesis that the privacy spectrum gives valuable insight into this question. In fact, this principle proves to be effective for getting insights on DP mechanisms as is demonstrated by our results on Section 8 and our prior work [2].[1]

In this work we define the *(differential) privacy spectrum* (DP spectrum) of a mechanism $\mathcal{M}$, denoted as $\delta_{\mathcal{M}}(\varepsilon)$,[2] to be the (function plotting the) minimum δ achievable for a given

1. We remark that [3] uses the notion of *privacy profile* for a quantity highly related to the privacy spectrum.

2. To avoid clutter, when the context is clear, we drop $\mathcal{M}$ from the notation, e.g., $\delta(\varepsilon)$ instead of $\delta_{\mathcal{M}}(\varepsilon)$.

ε. We propose an ML-based method for estimating the privacy spectrum of any given $\mathcal{M}$, while using $\mathcal{M}$ in a black-box manner. We prove accuracy guarantees of the estimated privacy spectrum using results from ML theory. The estimation error diminishes with the number of samples (runtime) that the estimator uses. We empirically demonstrate that the asymptotically-predicted behavior kicks-in already for a small number of samples. Our experiments demonstrate the potential of the proposed methodology for practical use, but also highlight the bottlenecks of the current proof-of-concept implementation, most relevantly, that estimating the spectrum for small values of δ, e.g., less than 10^{-7} requires a combination of excessive amount of storage and parallelism that is unavailable even in a modern supercomputer. We conjecture that the performance can be drastically improved by a combination of algorithmic, machine learning, and engineering optimizations, and view such improvements as an interesting research direction.

Our results in context. Before describing our contribution, we put our results in context with the state of the art in DP property-testing. (A more detailed discussion is included in Section 2.) Recent literature includes several instances of such DP testers, typically parameter estimators or detectors of privacy violations. Roughly speaking, the desirable properties of such methods are *accuracy, generality, and efficiency*, as discussed below. Our main impossibility result (Theorem 3) shows that a poly-time black-box (ε, δ)-DP estimator is impossible when the input domain is super-polynomial (or even unbounded, as is the case in the classical DP definition). Thus, the above work would typically discount on at least one of the above goals, e.g., on generality, by requiring white-box access to the mechanism [4], [5], [6], [7], or, on accuracy by testing a limited set (in order to preserve efficiency) of input databases to detect privacy violation [8], [9], [10] or to estimate (ε, δ)-DP [11]. In the following we discuss each of these properties, and how our Eureka framework approaches them.

Accuracy requires that the estimated DP-spectrum of $\mathcal{M}$ match its true DP-spectrum. There are two modes in which one can empirically analyze the DP spectrum. (1) *Verify* if a mechanism satisfies a given (ε, δ)-DP requirement. Typically one estimates *bounds* on the DP parameter(s) [8], [9], [10] to decide if the privacy is violated. The bounds can be loose, and so the verification may not be conclusive. (2) A stronger and more useful statement is to estimate the full DP-spectrum of the mechanism using *tight (upper and lower) bounds* on the privacy parameters. This is the task we tackle in this work. To our knowledge, the only other work which attempted such a tight estimation is the ADP-Estimator [11] which only applies to mechanisms with a small output space. (See Sec. 2 for a detailed comparison.)

The aforementioned prior work primarily offers heuristic empirical estimates of the privacy parameters. In contrast, we develop a framework for theoretical guarantees on the estimated privacy. Importantly, we demonstrate the concrete accuracy and efficiency of our estimator via proof-of-concept implementations in Sec. 8.

Generality mandates that the estimator should work for *any* mechanism. Our estimator achieves this by using the mechanism as a *black-box*, only interacting with the mechanism in an input/output manner. In contrast, a *white-box* estimator needs the (pseudo-code) of the mechanism whose privacy is to be estimated. An orthogonal feature of estimators regarding generality is whether they estimate only the ε parameter (aiming for the less flexible ε-DP) or, as we do in this work, estimate the full DP-spectrum which quantifies the ε-δ trade-off. The latter is more general, as ε-DP is the same as $(\varepsilon, 0)$-DP (setting $\delta = 0$).

Efficiency is necessary for the practicality of an estimator. Methodologies that exhaustively process the output space, such as [8], [9], [11], quickly become impractical, especially for large output spaces. Eureka puts forth several ideas to rectify this, e.g., by introducing the notion of *relative DP* (see below). As we discuss, at a technical level, relative DP allows us to bring the dependence of a mechanisms complexity on the universe of possible datasets from exponential down to linear. And on an intuitive level, the notion makes our methodology a fit for the prototypical scenarios of DP which include queries on large, static or slow evolving datasets, e.g., medical studies or census data processing. In fact, to our knowledge, our work proposes first tight, black-box, and theory-backed (ε, δ)-DP estimator that can handle mechanisms with a large (even uncountable) output space. Notwithstanding, the current instantiation of the Eureka framework, still has a efficiency bottleneck with respect to the size of δ as discussed above, which one needs to overcome to use it in practice. (Comprehensive comparisons of our estimator with existing methods are included in Section 2, *cf* Table 1.)

1.1. Our Contributions

We propose a general framework for constructing and analyzing black-box DP estimators. We analyze and benchmark a concrete instance of our framework. Our main insight is that a black-box DP-spectrum estimator can be re-cast as a specially-crafted classification problem, which can then be analyzed using ML techniques. In particular, given a data set and a (black-box) mechanism, we devise a new classification task whose optimal classifier can be directly linked to the DP-spectrum of the mechanism. Thus we can employ the results of this optimal classifier and estimate (theoretically and empirically) the DP-spectrum of the given mechanism. Concretely, using tools from statistical learning theory, we obtain tight bounds on the performance of this optimal classifier, and thus the DP-spectrum of the black-box mechanism. In the following, we elaborate on some of the main points and techniques and give pointers to the paper sections that include the detailed treatment.

Relative Differential Privacy (Section 4) First, we ask if it is even possible to estimate the (ε, δ)-DP-spectrum of an arbitrary mechanism. We prove that there are *no* efficient general black-box DP estimators, even if one allows an error probability β and an approximation factor α. As we show

in Theorem 3, for general input domains, an estimator with reasonable α and β is impossible. In fact, one can verify that the proof idea of our impossibility theorem applies even to relaxed definitions of DP from the literature, such as Renyi DP [12] and (Zero-)Concentrated DP [13], [14].

The core issue—one faced by all previous black-box DP estimators—is that by comparing every database with all its neighbors (databases resulting from removing or adding records), we get a universe infeasible to handle by an efficient estimator. The obvious way to circumvent the above issue is to limit the set of databases in a meaningful way. For example, a prototypical scenario where privacy is required (often by law) is research on medical data. It is well known that much of such research is done on just a handful of well-calibrated datasets. This leads us to propose relative DP, a relaxation of the DP definition.

Relative differential privacy (relative DP, Sec. 4.1) circumvents the above impossibility by limiting the input databases set: informally, an $(\varepsilon, \delta, \mathcal{T})$-relative DP mechanism is one which satisfies (ε, δ)-DP for databases in a given set $\mathcal{T}$. Observe that even with such a restriction, if we use the classical DP definition of neighboring databases, we can end up with an unmanageable universe of potential neighbors—since there may be an unlimited number of possible neighbors that *add* a database record/row. Instead, we use a refinement of the neighboring condition, namely for each database in $\mathcal{T}$ its neighbors are derived by *removing* any one record. An inspection of Def. 3 shows this yields an equivalent definition for DP, but now the number of neighbors for a database is simply the database's size, and thereby manageable for our estimator.

Further demonstrating the usefulness of our relaxation, we prove that relative DP has many of the desirable properties of DP that make it useful in a wide range of applications. In a nutshell, we prove in a sequence of results (Proposition 2-5) that relative DP is reasonably robust to adding new databases to the set $\mathcal{T}$, preserves privacy under mechanism composition, and is robust to post-processing.

(Relative) DP Estimator (Section 5, 6). Armed with relative DP, we devise and analyze a relative DP(-spectrum) estimator. To build intuition for our approach, we start (Sec. 5) with a toy case of estimating the smallest δ privacy parameter (given ε) for which two neighboring databases can satisfy inequality in the DP definition (Def. 3). We show how to convert the *risk* of the Bayes/optimal classifier to this minimum δ (Theorem 4). Intuitively, this conversion relies on writing δ as a statistical distance between two distributions, then using a well-known relationship between Bayes classifier risk and statistical distance. This leads to Lem. 1 for estimating the privacy for a single pair of neighboring databases. Then, we generalize our domain and extend $\mathcal{T}$ to be any polynomial-sized set of databases, using the scalability property (Prop. 2) of relative DP (Thm. 5). Note, our method works with any classifier whose risk converges to the Bayes/optimal risk.

In Sec. 6, we instantiate the general results above with the k-Nearest-Neighbor (kNN) [15] classifier. Utilizing the rich theory behind kNN (*cf.* Thm 2), we prove for this estimator the corresponding result for a single pair of neighboring databases: Cor. 2 as corollary to Lem. 1, and the general relative DP result: Cor. 3 as a corollary to Thm. 5.

Distributional Differential Privacy (Section 7) Assumptions on the data distribution can be used to replace the "relative" (to a specific $\mathcal{T}$) restriction of our treatment. This makes our framework applicable to "noiseless" versions of DP such as the well known *Distributional Differential Privacy* (DDP) notion [16]. In a nutshell, these notions take advantage of the inherent entropy in common datasets to reduce the amount of noise needed to achieve DP (see Section 3.1 for an overview.) We show that, under the assumption of independently distributed database rows, our relative DP estimator framework can be employed to estimate the DDP parameters of a mechanism. To our knowledge, this yields the *first black-box DDP estimator*. We believe that both the general paradigm and the estimator itself are of independent interest to ML research, where noiseless algorithms may achieve much better utility.

Empirical Validation and Benchmarks (see Section 8). Last but not least, we devise proof-of-concept implementations of our estimator, where we instantiated the classifier both with the well-studied and theory-backed kNN, and with a more efficient neural network. Our benchmarks and experiments serve the following three purposes:

1. *Validate our theory by showing the accuracy of our estimator:* **(i)** For mechanisms where we can analytically compute the (ε, δ) spectrum (e.g., Laplace, Gaussian, and exponential mechanisms[3]), our estimator output closely matches the analytically computed spectrum. This holds both for low-dimensional inputs (kNN-based estimator) and high-dimensional inputs (neural-network-based estimator). To our knowledge, our work is the first to enjoy this property. **(ii)** Our (kNN-based) estimator's concrete accuracy matches our proven theoretic (i.e., asymptotic) accuracy.[4]

We remark that Theorem 2 theoretically bounds the number of samples needed to guarantee a desired error for kNN.[5] As demonstated by our experiments, however, the above theoretical lower bound is overly pessimistic, and in practice, far fewer samples are sufficient. Our algorithm runs in $O(qn)$, where $q =$ number of neighboring databases tested (this is necessary dependency since a mechanism's behavior can vary drastically with database) and $n =$ number of samples. Achieving cryptographically small error in δ is feasible: 10^{-5} error needs just 2^{26} samples which takes $\sim$10 minutes on the textbook implementation of kNN.

3. Due to the nature of the exponential mechanism, we can only compute the (ε, δ) spectrum analytically for a given database, but this experiment serves as demonstration that our estimator works also for mechanisms with discrete output.

4. We mention that Antos *et al.* [17] proved there is no fixed finite sample-size beyond which one can universally bound the convergence rate of a Bayes risk estimator. Hence, empirical validation is the only way to demonstrate that the asymptotic predictions kick in for moderately-sized samples. (Such an empirical validation of asymptotic theory is common in both the cryptography/privacy and the ML literature).

5. Note that the dataset size and the number of samples the mechanism needs are distinct concepts. Domain experts do not need to be concerned about the latter, which is generated as part of the DP estimator process.

2. *Demonstrate applications of our estimator to*: **(i)** Compare the privacy spectrums of different mechanisms. **(ii)** Estimate the spectrum of more complex mechanisms for which we do not have tight analytically computed privacy bounds; for this we use the Sparse Vector Technique (SVT) mechanism for which we show that our estimator produces results that match the state of the art [11]. **(iii)** Test the correctness of mechanism implementations, a popular task studied in recent literature [4], [5], [6], [7], [8], [9], [10], [11]. We test buggy implementations of SVT, noisy histogram, and noisy max, that were used as benchmarks in the literature.
Lastly, **3.** *The first demonstration for estimating Distributional Differential Privacy (DDP).*

2. Related Work

Programming Language-based methods. This line of work uses language-based methods to automatically verify a mechanism's privacy level [4], [5], [6], [7]. These methods require *white-box* access to the mechanism. They are particularly useful in formally verifying if the implementation of some known mechanisms is correct or buggy. In particular, these estimators automatically search and infer proof of the DP property for the tested mechanism, hence the result (satisfying DP or not) is accurate if they do succeed. However, automated verification may fail. For example, [7] reports that LightDP [4] is unable to disprove faulty variants of PrivTree [19], because the variants have a probabilistic main loop with an unbounded number of iterations. Our work applies to general black-box mechanisms, e.g., proprietary software or heuristic attempts by ML researchers.

Probabilistic testing methods. This line of work uses statistical tools and is based on sampling the mechanism's inputs/outputs [8], [9], [10], [11], [18]. Specifically, they focus on lower-bounding the DP parameter of a mechanism—that is, asserting that the tested mechanism cannot achieve (beyond a) certain level of differential privacy. The core challenge is to find a witness of the DP violation for the given privacy parameters. StatDP [8] requires semi-black-box access to the mechanism to run the mechanism on input data without any noise. DP-Finder [9] requires the mechanism's algorithm (white-box access) to be differentiable, so that excludes common operations such as arbitrary loops or hash functions. This requirement considerably limits the class of mechanisms the method applies to, and excludes common DP techniques such as SVT [20] and Randomized Response [21]. DP-Sniper [10] and the more recent DPL [18] use the black-box approach and are general. DPL [18] improves upon DP-Sniper [10] by avoiding "event selection"—a major obstacle to finding a privacy violation witness. This is achieved via kernel density estimation. All these methods, including DP-Sniper and DPL test for ε-DP by only finding a *lower bound* of the privacy parameter ε on neighboring databases. In comparison, our work gives a tight characterization (i.e., *both* upper and lower bounds) on both the ε and δ privacy parameters.

ADP-Estimator [11] tests the (ε, δ)-DP property for a mechanism, and discusses the relationship between accuracy and number of samples required. While our goals align with [11], our approach is vastly different. ADP-Estimator empirically estimates the output distributions for a single pair of neighboring databases. In comparison, we develop a general framework that gives a formal treatment of the DP parameter-estimation problem and links it to the rich ML theory on classification algorithms, hence our method can derive a family of privacy estimators by plugging in different classifiers. In addition, the ADP-Estimator [11] is limited: by enumerating the tested mechanism's output space, their algorithm requires this space to be a finite (and small). In contrast, our estimator that uses the kNN classifier does not have such limitations. As further evidence of our method's advantage, we estimate the Gaussian mechanism (Section 8), which hadn't been reported by either DP-Sniper and DPL (they only consider ε-DP) or ADP-estimator (can't handle uncountable output space).

Machine Learning for DP The connection between machine learning and DP estimation has recently attracted attention in the ML/AI literature. A recent line of works [22], [23], [24], [25] investigated a connection between DP and empirically estimatable statistical distance. In a nutshell, these works bound the distinguishing advantage between distributions $\mathcal{M}(D)$ and $\mathcal{M}(D')$ (which relates to their statistical distance) for mechanism $\mathcal{M}$ and neighboring database D and D'. Specifically, given a (ε, δ)-DP mechanism, these works upper bound the statistical distance between $\mathcal{M}(D)$ and $\mathcal{M}(D')$, which lower bounds δ as a function of ε. In contrast, our results use a pair of carefully crafted distributions (not $M(D)$ and $M(D')$) which allows us to build an exact link between the DP-spectrum and a Bayes optimal risk. By then estimating this risk nonparametrically, we get tight upper and lower bounds on the achievable δ as a function of ε, accurately characterizing the entire DP-spectrum. Devising and analyzing these new distributions—and the connection to the DP-spectrum—is a key novelty here and can be seen as a non-trivial extension of Le Cam's (lower-only) bound [26]: we present equality rather than just a lower bound, which we use to tightly (upper and lower) bound the accuracy of our estimated δ.

Lastly, Gilbert and McMillan [27] discuss the sample complexity lower bound of verifying whether specific (ε, δ)-DP is satisfied. Their work is useful to answer what type of privacy parameter verification task is feasible. In contrast, our work devises a concrete method of *tightly estimating* privacy. To achieve this, we also develop sample complexity results that are orthogonal to [27].

3. Preliminaries

Due to the nature of our work lying in the intersection between privacy and ML, we provide the following preliminaries for both (1) privacy definitions and (2) background on ML classifiers, especially kNN.

	Access to $\mathcal{M}$	$\mathcal{M}$ with large output space	Accuracy	Methods
StatDP [8]	Semi-black-box	No	Lower bounds	Hypothesis testing
DP-Finder [9]	White-box	No	Lower bounds	Sampling and optimzation
DP-Sniper [10]	Black-box	Yes	Lower bounds	Classifier
DPL [18]	Black-box	Yes	Lower bounds	Kernel Density estimator
ADP-Estimator [11]	Black-box	No	Upper and lower bounds	Distribution estimator
Our Work	Black-box	Yes	Upper and lower bounds	Classifier (e.g., kNN)

TABLE 1: Summary of comparisons between our work and previous works.

3.1. Privacy Definitions

Informally, *differential privacy* (DP) [28] is defined via an experiment between a query party Q and a *curator* C, who has access to a database D. Q wishes to make a query on the database, and C wants to answer this query in a way that protects the privacy of any individual record. This property is achieved by C using a randomized algorithm, aka *mechanism*, to answer Q's queries, in a way that does not destroy accuracy (the outcome of the mechanism is not too far from the true answer to the query)—while respecting the privacy of any individual record $X \in D$ (informally, Q has only a small chance in telling whether or not X was used in answering the query). To make this formal, we state here the definition of DP (cf. [29] for an excellent treatment of DP and its properties.)

Definition 1 (**Mechanism**). *Let $\mathcal{U}$ be the set of all possible database records. Let $\mathcal{X} = \mathcal{U}^*$ be the set of all databases where each database row is from $\mathcal{U}$. Let $\mathcal{O}$ be the set of all possible output strings. Then a mechanism $\mathcal{M} := \mathcal{X} \mapsto \mathcal{O}$ is a (randomized) algorithm that takes as input a database from the input space $\mathcal{X}$, and produces an output from the output space $\mathcal{O}$.*

In DP, we are interested in whether our mechanism reveals information on individual database records/rows. Thus, we consider the output of our mechanism on pairs of databases called *neighbors*, where one neighbor contains a particular row, and the other does not.

Definition 2 (**Neighboring Databases**). *A pair of databases $D, D' \in \mathcal{X}$ is* neighboring*, denoted $D \simeq D'$ if D' can be obtained from D by removing one row*[6]

A mechanism is DP if its output given any D is similar to its output given any of D's neighbors.

Definition 3 (**Differential Privacy (DP)** [28]). *A mechanism $\mathcal{M} := \mathcal{X} \mapsto \mathcal{O}$ is (ε, δ)-differentially private if for all subset $\mathcal{S} \subseteq \mathcal{O}$ and for all neighboring databases $D \simeq D'$, $D, D' \in \mathcal{X}$:*

$$\Pr[\mathcal{M}(D) \in \mathcal{S}] \leqslant e^{\varepsilon} Pr[\mathcal{M}(D') \in \mathcal{S}] + \delta,$$

and

$$\Pr[\mathcal{M}(D') \in \mathcal{S}] \leqslant e^{\varepsilon} Pr[\mathcal{M}(D) \in \mathcal{S}] + \delta.$$

where the probability space is over the coin flips of the mechanism $\mathcal{M}$. If $\delta = 0$, we say that $\mathcal{M}$ is ε-differentially private.

Distributional Differential Privacy (DDP). The above DP definition is broadly used, but may be inapplicable in cases where utility degrades rapidly even with small noise, such as machine learning with deep networks, whose performance is sensitive to noise in the data. *Distributional differential privacy (DDP)* [16].[7] was suggested as an alternative to DP for such cases. The idea is that we might often be willing to make an assumption about the entropy (inherent randomness) of the database. Thus, instead of adding (too much) extra randomness/noise in the mechanism, we rely on this inherent randomness to achieve similar privacy guarantees as DP with less hit on the output's accuracy. Informally, DDP is defined via a similar inequality as DP, except instead of fixed pairs of neighboring databases, we consider databases as random variables (r.v.'s) from a distribution π, and condition each probability on a row i being set to some x or x'[8]. As the *first* demonstration of a DDP estimator, we consider the simple, special case of distributions π with independently distributed rows, where DDP can be reduced to the simpler definition below.

Definition 4 (Simplified DDP [33]). *Let Δ be a set of distributions on databases where each row is independently distributed. For any $\varepsilon > 0$ and $\delta > 0$, a mechanism $\mathcal{M}$ is $(\varepsilon, \delta, \Delta)$-DDP if for every $\pi \in \Delta$, $i \leqslant n$, $x, x' \in \mathcal{U}$, and $\mathcal{S} \subseteq \texttt{Range}(\mathcal{M})$, the following inequality holds.*

$$\Pr_{D\sim\pi}(\mathcal{M}(D) \in \mathcal{S} | D_i = x) \leqslant e^{\varepsilon} \Pr_{D\sim\pi}(\mathcal{M}(D) \in \mathcal{S} | D_i = x') + \delta,$$

3.2. Machine Learning Classifiers

Our treatment uses concepts and results from machine learning (ML) theory to construct our privacy estimator and prove (tight) bounds on its accuracy, i.e., how well it estimates optimal pairs (ε, δ) for the (D)DP definitions.

Let $\mathcal{O}$ denote the observation space, and let the label (or prediction) space be $\mathcal{Y} = \{0, 1\}$ (e.g., outputting 0 means the classifier predicts the observation is from one distribution and outputting 1 means the classifier predicts

6. See Sec. 1.1 for why we choose this definition of 'neighbors' instead of having D' be obtained either by removing or adding a row.

7. We focus here on DDP but we believe our approach applies also to alternative type of noiseless privacy [30], [31], [32].

8. For readers familiar with DDP, we note there is also auxiliary information Z, and a 'simulator' h. However, in the special case of $Z = \emptyset$ and independently distributed database rows, Def. 4 is shown [33] to be equivalent to DDP.

the other distribution). Let $\mathcal{P}$ be a joint distribution with the support of $\mathcal{O}\times\mathcal{Y}$, where $\mathcal{O}\times\mathcal{Y} := \{(o,b) : o \in \mathcal{O}, b \in \mathcal{Y}\}$ is a concatenation set. Let $\mathcal{I}(b,y)$ be the *inequality predicate*, i.e., the indicator function outputs 1 if b is not equal to y, otherwise 0.

A *classifier* $h : \mathcal{O} \mapsto \mathcal{Y}$ (also called a *classification algorithm*) is a function from the observation space $\mathcal{O}$ to the prediction space $\mathcal{Y}$. For every observation $o \in \mathcal{O}$, h outputs a bit $b \in \mathcal{Y}$ indicating that h predicts o has label b.

A *risk function* R is defined with respect to a distribution $\mathcal{P}$ on observables—in fact, it is easier to think of $\mathcal{P}$ as a joint distribution of pairs of the type (x,y) where x is an observation and y is its label. R takes a classifier h as input, and computes the probability that a sample drawn from $\mathcal{P}$ is mistakenly classified—i.e., assigned the wrong label—by h; equivalently, R computes the expectation of the above inequality predicate. Formally:

$$R(h) = \Pr_{(x,y)\sim\mathcal{P}}[\mathcal{I}(h(x),y)=1] = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{P}}[\mathcal{I}(h(x),y)].$$

We note that in a given application context, the risk $R(h)$ is typically impossible to compute, as the distribution $\mathcal{P}$ is unknown. However, viewing risk $R(h)$ as the expectation of the random variable $\mathcal{I}(h(x),y)$, allows us to derive a good estimator for it: the *testing risk* $\hat{R}_m(h)$ which is defined as the average on a set of independent samples $((x_1,y_1),\cdots,(x_m,y_m)) \sim \mathcal{P}^m$. (We make the sampling process $((x_1,y_1),\cdots,(x_m,y_m)) \sim \mathcal{P}^m$ implicit when it is clear from context). Formally:

$$\hat{R}_m(h) = \frac{1}{m}\sum_{i=1}^{m}\mathcal{I}(h(x_i),y_i),$$

In particular, a well-known result using Hoeffding's inequality allows us to gauge, up to an error probability γ, how close $\hat{R}_m(h)$ is to the true risk $R(h)$:

Theorem 1 (**Hoeffding's Inequality** [34])**.** *With probability* $1-\gamma$,

$$|\hat{R}_m(h) - R(h)| \leqslant \sqrt{\frac{1}{2m}\ln\frac{2}{\gamma}}.$$

Bayes (optimal) classifiers. A *Bayes (optimal) classifier* h^* with respect to $\mathcal{P}$ is a classifier that has the minimal risk $R(h^*)$ among all the classifiers (with respect to the same $\mathcal{P}$.)

The kNN Classifier Unfortunately, for the same reason we can not compute R—i.e., because $\mathcal{P}$ is typically unknown[9]—we can also not construct the Bayes classifier h^*. Nonetheless, ML theory provides us with several "reasonable" classifiers that achieve both good performance, and are close to optimal. One such classifier which is well understood and thoroughly studied in the field of pattern recognition is the *k-Nearest Neighbor (kNN) classifier*—which we use in our paper as a concrete instantiation of our framework. To construct a kNN classifier $h^{\mathtt{NN}}_{k,n}$ with n samples, we simply sample and store n training points $((x_1,y_1),\cdots,(x_n,y_n)) \sim \mathcal{P}^n$. To predict the label of an observation $o \in \mathcal{O}$, $h^{\mathtt{NN}}_{k,n}$ returns the label taking a majority vote of the class labels of its k nearest neighbors (according to the distance function defined on the space) in the stored training points:

$$h^{\mathtt{NN}}_{k,n}(o) = \left\lfloor \frac{1}{k}\sum_{i\in[k]} b_i \right\rceil,$$

where b_i is the label of the i-th nearest neighbor of o, and $\lfloor\cdot\rceil$ is an operator rounding to nearest integer.

The following convergence result for kNN gauges how close the true risk $R(h^{\mathtt{NN}}_{k,n})$ of the kNN classifier $h^{\mathtt{NN}}_{k,n}$ is to the risk of the optimal classifier, $R(h^*)$.

Theorem 2 (**Convergence of k-Nearest Neighbor Classifier** [15])**.** *Let $\mathcal{P}$ be a joint distribution with support $\mathcal{O}\times\mathcal{Y}$. If the conditional distribution $\mathcal{P}|\mathcal{Y}$ has a density, $\mathcal{O} \subseteq \mathsf{R}^d$, and $k=\sqrt{n}$, then for every $\alpha > 0$ there is an n_0 such that for $n > n_0$,*

$$\Pr[|R(h^{\mathtt{NN}}_{k,n}) - R(h^*)| > \alpha] \leqslant 2e^{-n\alpha^2/(72c_d^2)},$$

where c_d[10] is the minimal number of cones centered at the origin of angle $\pi/6$ that cover R^d. Note that if the number of dimensions d is constant, then c_d is also a constant.

4. Privacy Estimation and Relative DP

In this section, we describe the problem of (black-box) privacy estimation, and its limitations, which will motivate our new notion of *relative DP*.

At a high level, a privacy estimator is an algorithm which, given a mechanism $\mathcal{M}$ and an ε value, outputs the optimal (smallest) δ for which $\mathcal{M}$ is (ε,δ)-DP (symmetrically, it can also be given δ and be asked to estimate ε). As we will show, the above task is impossible without relaxing the requirements on the accuracy of the estimator. More concretely, we say that an estimator achieves accuracy bounds (α,β) when, on inputs these two parameters, its output δ is at most α-far from the correct answer with probability at least $1-\beta$.

Below, we first define the notion of *optimal* δ given any ε and mechanism $\mathcal{M}$. Note that this optimal δ is a point on the DP-spectrum (curve) discussed in the introduction. We also define the quantity $\delta_{D,D'}$ which is the optimal δ *with respect to a single, fixed pair of (neighboring) databases* D, D'. Looking ahead in the next section, we will first tackle the easier problem of estimating $\delta_{D,D'}$ (Section 5.1), before tackling the harder problem of estimating δ itself (Section 5.2).

Definition 5 (**Optimal** δ)**.** *The privacy parameter $\delta_{D,D'}$ is optimal (minimal) with respect to the tuple $(\mathcal{M},D,D',\varepsilon)$ if*

$$\delta_{D,D'} = \max(\max_{\mathcal{S}\subseteq\mathcal{O}} \Pr[\mathcal{M}(D)\in\mathcal{S}] - e^{\varepsilon}\Pr[\mathcal{M}(D')\in\mathcal{S}], 0).$$

The privacy parameter δ is optimal (minimal) with respect to the tuple $(\mathcal{M},\varepsilon)$ if

$$\delta = \max_{D\simeq D'}\left\{\max(\delta_{D,D'},\delta_{D',D})\right\}.$$

9. In a typical ML classification experiment, one is able to observe values sampled from $\mathcal{P}$ but does not know the actual distribution.

10. By Lemma 5.5 of [15], c_d satisfies $c_d \leqslant (1+2/\sqrt{2-\sqrt{3}})^d - 1$.

Then, we define a (perfect) DP estimator, which, given any mechanism $\mathcal{M} \in \mathcal{C}$ from a set of mechanisms $\mathcal{C}$ and one of the privacy parameters ε, outputs the optimal δ such that $\mathcal{M}$ is (ε, δ)-DP.

Definition 6 (**Perfect DP Estimator**). *An algorithm is a Perfect DP Estimator for $\mathcal{C}$, if for every $\mathcal{M} \in \mathcal{C}$ and $\varepsilon \in \mathsf{R}_{\geqslant 0}$, with black-box access to $\mathcal{M}$, the algorithm outputs the optimal δ with respect to the tuple $(\mathcal{M}, \varepsilon)$.*

Unfortunately, a perfect DP estimator does not exist. In fact, we can show something even stronger—even an approximate version of a DP estimator (Def. 8) still does not exist (Theorem 3). Intuitively, this is because a general estimator would need to test the DP property for all pairs of databases—an impossible task for a polynomial-time algorithm if the number of databases in the mechanism's domain is super-polynomial. The proof of the theorem follows the above intuition and can be found in Appendix B.

Definition 7 (α-**tight bound**). *The estimate $\delta'_{D,D'}$ is a α-tight bound with respect to $(\mathcal{M}, D, D', \varepsilon)$ if*

$$|\delta'_{D,D'} - \delta_{D,D'}| \leqslant \alpha,$$

where $\delta_{D,D'}$ is optimal with respect to $(\mathcal{M}, D, D', \varepsilon)$.

Similarly, we say δ' is a α-tight bound with respect to $(\mathcal{M}, \varepsilon)$ if

$$|\delta' - \delta| \leqslant \alpha,$$

where δ is optimal with respect to $(\mathcal{M}, \varepsilon)$.

Definition 8 (**Approximate DP Estimator**). *An algorithm is a (α, β)-Approximate DP Estimator for $\mathcal{C}$, if for every $\mathcal{M} \in \mathcal{C}$ and $\varepsilon \in \mathsf{R}_{\geqslant 0}$, with black-box access to $\mathcal{M}$, with probability at least $1 - \beta$, it provides α-tight bound with respect to the tuple $(\mathcal{M}, \varepsilon)$, where $\alpha, \beta \in [0, 1)$.*

Theorem 3 (Impossibility of even an *approximate* DP estimator, Proof in Appendix B). *Let $\alpha \in [0, \frac{1}{2})$ and $\beta \geqslant \frac{1}{2} + \nu(n)$, where ν is a non-negligible function. Let $\mathcal{C} = \{0,1\}^n \mapsto \mathcal{O}$ be the set of $\mathrm{poly}(n)$-time mechanisms. There doesn't exist a $\mathrm{poly}(n)$-time (α, β)-Approximate DP Estimator for $\mathcal{C}$.*

Remark 1 (On the generality of the impossibility result). *One can verify that the above impossibility also applies to common relaxations of DP from the literature, such as Renyi DP [12] and (Zero-) Concentrated DP [13], [14]. Intuitively, the core reason for Theorem 3—the need to test all pairs of neighbors to in general have accurate privacy estimates—applies also to the above variants. This points to the idea that in order to circumvent our impossibility, it seems necessary to bound the size of the mechanism's input space, which motivates the* relative-DP *relaxation detailed in the following.*

4.1. Relative Differential Privacy

In view of the impossibility stated in Theorem 3, we ask: "Is there a meaningful/useful relaxation to differential privacy that allows us to circumvent it?" To answer the above question in affirmative, we introduce *relative differential privacy*, which we believe is both minimal (in terms of intuitive distance from DP) and useful in typical privacy-requiring applications, such as medical research and census-data statistics release, as discussed below and on Section 1. Relative DP considers the privacy of a mechanism *relative to a set of databases*. Informally, a mechanism is $(\varepsilon, \delta, \mathcal{T})$-relative DP if on domain restricted to $\mathcal{T}$, the mechanism is (ε, δ)-DP.

Recall, we defined "neighboring" (Def. 2) as "remove one row" rather than "remove-or-add one row", so that the number of neighbors of a database does not depend on the domain size of each database row (i.e., the number of possible "add-one-row" neighbors). This modification did not change the original DP definition, but allows our Thm 5 to circumvent impossibility Thm 3 for superpolynomial-size domains in our *relative DP* definition. Indeed, using relative DP instead of (classical) DP reduces the complexity dependence of a privacy estimator on the number of possible records in our dataset from exponential down to linear. We note in passing that the use of relative DP also allows them to focus their estimation on the (classes) of databases relevant for their experiment, which, we believe, makes the task more approachable for domain experts, e.g. medical researchers, who are only interested on specific classes (or even specific sets) of data.

Definition 9 ($\varepsilon, \delta, \mathcal{T}$-**relative Differential Privacy**). *A mechanism $\mathcal{M} := \mathcal{X} \mapsto \mathcal{O}$ is $(\varepsilon, \delta, \mathcal{T})$-relative differentially private if $\mathcal{T} \subseteq \mathcal{X}$ and for all subset $\mathcal{S} \subseteq \mathcal{O}$ and all neighboring databases $D \simeq D' : D \in \mathcal{T}$:*

$$\Pr[\mathcal{M}(D) \in \mathcal{S}] \leqslant e^{\varepsilon} Pr[\mathcal{M}(D') \in \mathcal{S}] + \delta,$$

and

$$\Pr[\mathcal{M}(D') \in \mathcal{S}] \leqslant e^{\varepsilon} Pr[\mathcal{M}(D) \in \mathcal{S}] + \delta.$$

where the probability space is over the coin flips of mechanism $\mathcal{M}$.

Remark 2 (Relative DP vs. *concrete DP*). *An arguably simpler-to-define DP-relaxation that circumvents our impossibility would be to limit the domain in the classical DP definition to a given finite set $\mathcal{X}$ (for the sake of this explanation we refer to this relaxation as* concrete *differential privacy)*[11]*, and require that the mechanism gives indistinguishable answers for any two databases in $\mathcal{X}$. This would in fact yield a more general definition—relative DP is an instance of the above when the $\mathcal{X}$ includes $\mathcal{T}$ and the neighbours of databases in $\mathcal{T}$. However, as discussed below, this would result in a less intuitive definition which would be harder to use by non-domain-experts. Indeed, a definition which specifies the domain $\mathcal{X}$ in such an arbitrary manner leaves it to the "consumer" of the definition (e.g., non-privacy-expert) to instantiate it with the right $\mathcal{X}$. Instead, relative DP removes this responsibility from the consumer and automatically specifies a natural set (of neighbors) for any given set of databases, so that the intuitive privacy*

11. In classical DP, $\mathcal{X}$ is the set of all databases and their neighbors.

requirement (i.e., hiding any record's participation in the mechanisms's output) is guaranteed.

Adding to the above, we next show that relative DP satisfies a very useful notion of $\mathcal{T}$-scalability (Prop. 2), which is not satisfied if we simply limit domain $\mathcal{X}$ (as in concrete DP), and enjoys several useful properties of DP: composition (Prop. 3, and 4) and post-processing (Prop. 5)) (see Appendix B for proofs). In fact, as we show next, relative DP and DP are the same, if $\mathcal{T}$ is defined as the (DP) domain of the mechanism.

Proposition 1. *If the mechanism $\mathcal{M}$ is $(\varepsilon, \delta, \mathcal{T})$-relative differentially private and $\mathcal{T} = \mathcal{X}$, then the mechanism $\mathcal{M}$ is (ε, δ)-differentially private.*

One might be worried that by providing such a relative version of DP, we might be creating a privacy notion that melts down once new databases are added to the mix. The following proposition shows that this is not the case for relative DP, as long as the mechanism behaves well on the new database.

Proposition 2. *[$\mathcal{T}$ **Scalable**] If the mechanism $\mathcal{M}$ is $(\varepsilon_1, \delta_1, \mathcal{T}_1)$-relative differentially private, $\cdots$, and $(\varepsilon_k, \delta_k, \mathcal{T}_k)$-relative differentially private, then the mechanism is also $\left(\max_{i\in[k]} \varepsilon_i, \max_{i\in[k]} \delta_i, \bigcup_{i\in[k]} \mathcal{T}_i\right)$- relative DP.*

Relative DP also enjoys the same convenient guarantees as DP: parallel composition, sequential composition, as well as post-processing.

Proposition 3. *[**Parallel Composition**] Let $\mathcal{T}_1 \times \mathcal{T}_2$ be the concatenation of set $\mathcal{T}_1$ and $\mathcal{T}_2$, that is, $\mathcal{T}_1 \times \mathcal{T}_2 = \{(D_1, D_2) : D_1 \in \mathcal{T}_1 \wedge D_2 \in \mathcal{T}_2\}$. If $\mathcal{M}_1, \cdots, \mathcal{M}_k$ are k mechanisms, where $\mathcal{M}_i$ satisfies $(\varepsilon_i, \delta_i, \mathcal{T}_i)$-relative differential privacy, then the mechanism $\mathcal{M}$ taking database $(D_1, \cdots, D_k) \in \mathcal{T}_1 \times \cdots \times \mathcal{T}_k$ as inputs and outputting $(\mathcal{M}_1(D_1), \cdots, \mathcal{M}_k(D_k))$ is $\left(\max_{i\in[k]} \varepsilon_i, \max_{i\in[k]} \delta_i, \mathcal{T}_1 \times \cdots \times \mathcal{T}_k\right)$-relative DP.*

Proposition 4. *[**Sequential Composition**] If $\mathcal{M}_1, \cdots, \mathcal{M}_k$ are k mechanisms, where $\mathcal{M}_i$ satisfies $(\varepsilon_i, \delta_i, \mathcal{T})$- relative differentially privacy, then the mechanism $\mathcal{M} := (\mathcal{M}_1, \cdots, \mathcal{M}_k)$ is $\left(\sum_{i\in[k]} \varepsilon_i, \sum_{i\in[k]} \delta_i, \mathcal{T}\right)$- relative DP.*

Proposition 5. *[**Post-processing**] If $\mathcal{M}_1$ is a mechanism that satisfies $(\varepsilon, \delta, \mathcal{T})$-relative differential privacy, then for any (randomized) algorithm f, the mechanism $\mathcal{M} := f(\mathcal{M}_1)$ is $(\varepsilon, \delta, \mathcal{T})$-relative differentially private.*

5. (Relative) DP Estimator

In this section, we define and analyze our (relative) privacy estimator, which can be instantiated with any classifier. Looking ahead, in Sec. 6, we will analyze our estimator instantiated with the well-studied k-nearest neighbor (kNN) classifier.

As discussed in the introduction, we start (in Section 5.1) with estimating the optimal delta for one pair of (neighboring) databases, i.e., $\delta_{D,D'}$ (Def. 5). Although this is clearly not particularly relevant for a general privacy definition, it allows us to introduce our main ideas (namely, constructing a DP estimator out of classifiers—see Sec. 5.1.2), and allows us a smooth transition to our general estimator which is described and analyzed in Section 5.2.

5.1. Estimating δ for a pair of databases

As the first step in defining our privacy estimator, we narrow the definition of a privacy estimator to define a privacy estimator for a single pair of neighboring databases. We construct a class of concrete privacy estimator algorithms $\mathcal{A}^{\mathsf{B}}_{\mathcal{C}}$ by relating the privacy parameter δ to the *risk* (or error) of a classification algorithm B (Theorem 4). We apply Thm. 4 to prove Lemma 1, which constructs a $\delta_{D,D'}$-estimator for a pair of neighboring databases.

Our results in this section show that, despite the impossibility of general DP estimators and the lack of tight bounds in previous work, it is indeed possible to construct relative DP estimators with tight accuracy bounds. In the next subsection, we will extend algorithm $\mathcal{A}^{\mathsf{B}}_{\mathcal{C}}$ of this section to construct a privacy estimator for any $(\varepsilon, \delta, \mathcal{T})$-relative DP mechanism.

5.1.1. Privacy Estimator for a Pair of Databases. First, we define a perfect δ estimator for a pair of databases. Informally, this estimator must always output the optimal δ (see Def. 5).

Definition 10 (Perfect δ-Estimator for a Pair of Databases). *An algorithm is a Perfect δ-Estimator for a Pair of Databases for $\mathcal{C}$ if for every $\mathcal{M} \in \mathcal{C}$, a pair of databases D, D' and $\varepsilon \in \mathsf{R}_{\geqslant 0}$, with black-box access to $\mathcal{M}$, the algorithm outputs the optimal $\delta_{D,D'}$ with respect to the tuple $(\mathcal{M}, D, D', \varepsilon)$.*

Unfortunately, a perfect estimator, even for just a pair of (neighboring) databases, does not exist—by Theorem 4, a perfect estimator would imply the existence of an optimal classifier achievable with limited training samples. Thus, we define below an approximate estimator Def. 11, with similar approximation parameters α and β as for the approximate DP privacy estimator Def. 8.

Definition 11 (Approximate δ-Estimator for a Pair of Databases). *An algorithm is a (α, β)-Approximate δ-Estimator for a Pair of Databases for $\mathcal{C}$ if for every $\mathcal{M} \in \mathcal{C}$, a pair of databases D, D' and $\varepsilon \in \mathsf{R}_{\geqslant 0}$, with black-box access to $\mathcal{M}$, with probability at least $1 - \beta$, it provides α-tight bound with respect to the tuple $(\mathcal{M}, D, D', \varepsilon)$, where $\alpha, \beta \in [0, 1)$.*

5.1.2. Relating Privacy Parameter δ to Risk of the Bayes Classifier. We now turn to construct the approximate privacy estimator with respect to a pair of databases (defined in Def. 11). The basis of our estimator is a connection between

the definition of DP and the *risk* of a Bayes Classifier, described in Theorem 4 below.

For a mechanism $\mathcal{M}$, a database D, and privacy parameter ε, let $\left[\mathcal{M}(D)\right]_\varepsilon$ denote the random variable obtained by tossing a biased coin c where $\Pr[c = 1] = e^{-\varepsilon}$, and receiving value $\mathcal{M}(D)$ if $c = 1$ or receiving value $\perp$ (a null value not in the range of $\mathcal{M}$) if $c = 0$.

Definition 12 (The distribution $\mathcal{P}_{(\mathcal{M},D,D',\varepsilon)}$)**.** *Let $\mathcal{P}_{(\mathcal{M},D,D',\varepsilon)}$ denote the distribution of a random variable, which is obtained by tossing a* fair *coin b, and receiving tuple $(\mathcal{M}(D'), 1)$ if $b = 1$ or receiving value $(\left[\mathcal{M}(D)\right]_\varepsilon, 0)$ otherwise.*

The proof of the theorem below (App. B) is based on the fact that δ in (ε, δ)-(relative) DP can be re-written in terms of a *statistical distance*[12] between two random variables. The difference between the DP definition and statistical distance is that in DP, one of the probabilities is scaled by e^ε. This means we can re-write $\delta_{D,D'}$ in terms of the statistical distance between two r.v.'s $\mathcal{M}(D')$ and $\left[\mathcal{M}(D)\right]_\varepsilon$ (which, intuitively, "scales" the distribution of $\mathcal{M}(D)$ by $1/e^\varepsilon$). Then, the theorem follows from the connection between statistical distance and the accuracy (or risk) of the optimal (or Bayes) classifier.

Theorem 4 (Mechanism Privacy as Bayes Classifier Risk, Proof in Appendix B)**.** *Let $h^*_{D,D'}$ be the Bayes classifier for $\mathcal{P}_{(\mathcal{M},D,D',\varepsilon)}$ (Def. 12, abbreviated as $\mathcal{P}$ below). The optimal delta $\delta_{D,D'}$ with respect to the tuple $(\mathcal{M}, D, D', \varepsilon)$ satisfies the following equality*

$$\delta_{D,D'} = \max\left(1 - 2e^\varepsilon R(h^*_{D,D'}), 0\right).$$

Corollary 1 states the relationship between finding the optimal δ and the risk of optimal Bayes classifiers. This implies that accuracy of *any* ML-based DP estimator is inherently tied to the accuracy of its underlying classifier.

Corollary 1. *The optimal δ with respect to the tuple $(\mathcal{M}, \varepsilon)$ satisfies the equality*

$$\delta = \max_{D \simeq D'} \left\{\max(1 - 2e^\varepsilon R(h^*_{D,D'}), 1 - 2e^\varepsilon R(h^*_{D',D}), 0)\right\}.$$

5.1.3. Privacy Estimator for Neighboring Databases with Tight Accuracy Bounds. In this section, we take advantage of the connection between DP and the risk of the Bayes classifier (Theorem 4), to construct an approximate DP estimator for a single pair of databases (see Def. 11). Our algorithm $\mathcal{A}^{\mathsf{B}}_{\mathcal{C}}$, Fig. 1, is parameterized by any classifier B, and generates a privacy estimate via the computed risk of this classifier.

Lemma 1 (δ-Estimator for a Pair of Databases Given Any Classifier, Proof in Appendix B)**.** *Let $h^*_{D,D'}$ be the Bayes classifier for $\mathcal{P}$ and h^{B}_n a classifier for $\mathcal{P}$ produced by classification algorithm B with n samples. h^{B}_n is consistent with $h^*_{D,D'}$: there is a function $g(\mathcal{X}, n, \beta)$ of input space $\mathcal{X}$, sample size n and $\beta \in (0, 1)$ such that $|R(h^{\mathsf{B}}_n) - R(h^*_{D,D'})| \leqslant g(\mathcal{X}, n, \beta)$.*

12. Statistical distance between two r.v. X, Y is defined as $\Delta(X, Y) = \max_{\mathcal{S}} |\Pr(X \in \mathcal{S}) - \Pr(Y \in \mathcal{S})|$.

Then, the algorithm $\mathcal{A}^{\mathsf{B}}_{\mathcal{C}}$ with n samples, shown in Figure 1 , is a (α, β)-Approximate δ-Estimator for a Pair of Databases for $\mathcal{C}$, for any $\alpha = 2e^\varepsilon g(\mathcal{X}, n/2, \beta/2) + 2e^\varepsilon\sqrt{\ln(4/\beta)/n}$, $\beta \in (0, 1)$.

In other words, the estimator's error diminishes quickly if the classifier achieves close-to-optimal risk quickly as the number of samples n grows (i.e., g diminishes quickly). The lemma also shows that the error α grows proportionally with e^ε; however, since usually ε is a small constant < 1, the effect of ε is small.

5.2. Estimating Approximate Relative DP

In this section, we extend our algorithm from our previous section, to construct a privacy estimator for relative DP (Def. 9). We begin with a formal definition for a relative DP estimator with tight bounds (Def. 14). Then, we present our privacy estimator which builds upon algorithm $\mathcal{A}^{\mathsf{B}}_{\mathcal{C}}$ from Section 5.1.3. Given any classification algorithm B, our privacy estimator $\mathcal{A}^{B}_{\mathcal{C},t}$ outputs the privacy parameter for any mechanism in class $\mathcal{C}$ and set of databases of size t.

5.2.1. Our Approximate Relative DP Estimator. Before describing our DP estimator, we first define the guarantees such a (α, β)-approximate relative DP estimator should satisfy. Intuitively, these are the same as for an approximate DP estimator, except we restrict the domain of our mechanism to the set $\mathcal{T}$, relative to which we define privacy.

Definition 13. *Let $\mathcal{T} \subseteq \mathcal{X}$ be a set of databases. We say the privacy parameter $\delta_{\mathcal{T}}$ is optimal with respect to $(\mathcal{M}, \mathcal{T}, \varepsilon)$, if*

$$\delta_{\mathcal{T}} = \max_{\substack{D \in \mathcal{T}: \\ D \simeq D'}} \left\{\max(\delta_{D,D'}, \delta_{D',D})\right\},$$

where $\delta_{D,D'}$ is optimal with respect to $(\mathcal{M}, D, D', \varepsilon)$. We say $\delta'_{\mathcal{T}}$ is an α-tight bound with respect to $(\mathcal{M}, \mathcal{T}, \varepsilon)$, if $|\delta'_{\mathcal{T}} - \delta_{\mathcal{T}}| \leqslant \alpha$.

Definition 14 (Approximate Relative DP Estimator). *An algorithm is a (α, β)-*Approximate Relative DP Estimator *for $\mathcal{C}$ if for every $\mathcal{M} \in \mathcal{C}$, set $\mathcal{T} \subseteq \mathcal{X}$ such that the size of $|\mathcal{T}|$ is finite and $\varepsilon \in \mathsf{R}_{\geqslant 0}$, with black-box access to $\mathcal{M}$ with probability at least $1 - \beta$, it provides α-tight bound with respect to the tuple $(\mathcal{M}, \mathcal{T}, \varepsilon)$ for any $\alpha, \beta \in [0, 1)$.*

We are now ready to formally define and analyze our Algorithm, denoted as $\mathcal{A}^{B}_{\mathcal{C},t}$ (see Fig. 2 for a detailed description). $\mathcal{A}^{B}_{\mathcal{C},t}$ uses our estimator for pairs of neighboring databases (see Fig. 1) and runs it for all neighbors for set $\mathcal{T}$. Intuitively, by union bound, our accuracy degrades multiplicatively with the total number of neighboring databases.

Theorem 5 ((Relative) DP Estimator Given Any Classifier, Proof in Appendix B)**.** *Let $h^*_{D,D'}$ be the Bayes classifier for $\mathcal{P}$ and h^{B}_n a classifier for $\mathcal{P}$ produced by classification algorithm B with n samples. Let h^{B}_n be consistent with $h^*_{D,D'}$: there is a function $g(\mathcal{X}, n, \beta)$ of input space $\mathcal{X}$, sample size n and $\beta \in (0, 1)$ such that $|R(h^{\mathsf{B}}_n) - R(h^*_{D,D'})| \leqslant$*

Input: A binary classification algorithm B with n samples. A mechanism $\mathcal{M} \in \mathcal{C}$, a pair of databases $D, D' \in \mathcal{X}$, privacy parameter $\varepsilon \in \mathsf{R}_{\geqslant 0}$.
Output: $\delta'_{D,D'}$, the estimate of the optimal delta $\delta_{D,D'}$ with respect to the tuple $(\mathcal{M}, D, D', \varepsilon)$.
Recall $\mathcal{P}_{(\mathcal{M},D,D',\varepsilon)}$ (Def. 12, abbreviated below as $\mathcal{P}$) denotes the distribution of a random variable, which is obtained by tossing a fair coin b, and receiving tuple $(\mathcal{M}(D'), 1)$ if $b = 1$ or receiving value $(\left[\mathcal{M}(D)\right]_\varepsilon, 0)$ [a] otherwise.

1) Initialize $n_1 \leftarrow n/2$, $n_2 \leftarrow n/2$, and $r \leftarrow 0$.
2) Sample n_1 training points $(o_1, b_1), \cdots, (o_{n_1}, b_{n_1})$ according to joint distribution $\mathcal{P}$.
3) Taking the n_1 training points as inputs, classification algorithm B outputs a classifier $h^{\mathtt{B}}_{n_1}$.
4) Repeat the process n_2 times: ▷ Estimate risk function of classifier $h^{\mathtt{B}}_{n_1}$ with n_2 testing samples.
   a) Sample a testing point (o, b) according to joint distribution $\mathcal{P}$.
   b) Predict the sample o's label using the trained classifier: $b' = h^{\mathtt{B}}_{n_1}(o)$. If $b' \neq b$, $r \leftarrow r + 1/n_2$.
5) Output $\delta'_{D,D'} \leftarrow \max(1 - 2e^{\varepsilon} r, 0)$.

a. Recall $\left[\mathcal{M}(D)\right]_\varepsilon$ is a distribution for tossing a coin c where $\Pr[c = 1] = e^{-\varepsilon}$, outputting $\mathcal{M}(D)$ if $c = 1$ or $\perp$ (a null value) otherwise.

Figure 1: $\mathcal{A}^{\mathtt{B}}_{\mathcal{C}}$, an algorithm for estimating the optimal delta with respect to the tuple $(\mathcal{M}, D, D', \varepsilon)$

$g(\mathcal{X}, n, \beta)$. *Let $\mathcal{T} \subseteq \mathcal{X}$ be any set of databases in relative DP, $|\mathcal{T}| \leqslant t$; m be the maximum number of rows in a database.*

Then, the algorithm $\mathcal{A}^B_{\mathcal{C},t}$, shown in Figure 2, is a (α, β)-Approximate Relative DP Estimator for $\mathcal{C}$, where $\alpha = 2e^{\varepsilon} g(\mathcal{X}, n/2, \beta/4tm) + 2e^{\varepsilon}\sqrt{\ln(8tm/\beta)/n}$, $\beta \in (0, 1)$.

We observe that Theorem 5 implies that the accuracy of our (black-box) estimator methodology is inherently dependent on $|\mathcal{T}|$. (In fact, Theorem 3 implies that the problem is intractable for superpolynomial $|\mathcal{T}|$.) To our knowledge, such a dependence is in fact implicit in all previous works on privacy estimators [8], [9], [11].

It is also worth noting that it follows from Proposition 2 that modifying a subset of databases in $\mathcal{T}$ does not require re-running the estimator on the whole set $\mathcal{T}$. Instead we only need to run it on the subset $\mathcal{T}' \subset \mathcal{T}$ of modified databases and adopt the larger δ (from the ones corresponding to sets $\mathcal{T}$ and $\mathcal{T}'$). This overhead becomes even less relevant for our motivating scenarios of statistics on medical or census data, where the databases are mostly static and updated rarely (new records are batched).

6. (Relative) DP Estimator based on kNN

Our results in Section 5 prove the theoretical accuracy bounds of our estimator, with respect to *any* classifier. In this section, we demonstrate how to apply our general results to the case when the classifier is kNN. Informally, the corollaries below hold since kNN satisfies the consistency requirement in Lemma 1 and Theorem 5 via Theorem 2 [15]. The proofs can be found in Appendix B.

Note that since Theorem 2 is asymptotic, our tight accuracy bounds are asymptotic as well.

The first corollary establishes the accuracy of our kNN-based estimator for a single pair of databases (i.e., some $D \in \mathcal{T}$ and one of its neighbors). The statement in Corollary 2 uses as black box Thm 11.1 in Devroye et al. [15] to give convergence which is asymptotic (in n). To get more concrete convergence rates such as fixing n_0 would require further assumptions beyond just the existence of a density. Note that such assumptions are necessary due to the impossibility result of Antos *et al.* [17] which implies the non-existence of a Bayes error estimate with universal convergence rate. Indeed our Theorem 4 links DP-estimation to Bayes error rates, so Antos *et al.* [17] implies there is no DP-estimator with a universal convergence rate. This means *any* DP-estimator with provable convergence rate must makes such assumptions.

Corollary 2 (δ-Estimator for a Pair of Databases Given kNN Classifier, Proof in Appendix B)**.** *Consider the set of mechanisms $\mathcal{C} = \mathcal{X} \mapsto \mathsf{R}^d$ whose output distributions have a density.* kNN *is the kNN classification algorithm with n samples where $k = \sqrt{n}$. Then there exists a n_0 such that for all $n > n_0$, the algorithm $\mathcal{A}^{kNN}_{\mathcal{C}}$ (Fig. 1) is a (α, β)-Approximate δ-Estimator for a Pair of Databases for $\mathcal{C}$, for any $\alpha = 24e^{\varepsilon} c_d \sqrt{\ln(4\beta)/n} + 2e^{\varepsilon}\sqrt{\ln(4/\beta)/n}$, $\beta \in (0, 1)$, where $c_d \leqslant (1 + 2/\sqrt{2 - \sqrt{3}})^d - 1 \leqslant 4.86371^d$ (Lemma 5.5, [15]).*

Below, Corollary 3 shows the accuracy of our privacy estimator based on the kNN classifier.

Corollary 3 ((α, β)-Approximate Relative DP Estimator, using kNN, Proof in Appendix B)**.** *Consider the set of mechanisms $\mathcal{C} = \mathcal{U}^m \mapsto \mathsf{R}^d$ whose output distribution has a density. Let $\mathcal{T} \subseteq \mathcal{X}$ be any set of databases in relative DP, $|\mathcal{T}| \leqslant t$. Let the algorithm B be $\mathcal{A}^{\mathtt{kNN}}_{\mathcal{C}}$ with n samples, shown in Figure 1. Then there exists a n_0 such that for all $n > n_0$, the algorithm $\mathcal{A}^B_{\mathcal{C},t}$ (Fig. 2) is a (α, β)-Approximate Relative DP Estimator for $\mathcal{C}$, where* $\alpha = 24e^{\varepsilon} c_d \sqrt{\ln(8tm/\beta)/n} + 2e^{\varepsilon}\sqrt{\ln(8tm/\beta)/n}$, $\beta \in (0, 1)$.

From Corollary 3 we see that using the kNN classifier, our privacy estimator does the best when dimension is very low (as error increases exponentially with dimension), and that sample size n reduces error by a $\sqrt{n}$ factor. The estimator error also increases by $\sqrt{\log(tm)}$, where tm computes the maximum number of neighbors of $\mathcal{T}$ on which to test the DP inequality (Def. 3).

Input: An algorithm B with n samples, which estimates the optimal $\delta_{\mathcal{T}}$ with respect to the tuple $(\mathcal{M}, D, D', \varepsilon)$ for mechanism family C. A mechanism $\mathcal{M} \in \mathcal{C}$, a set of databases $\mathcal{T}$, privacy parameter $\varepsilon \in \mathsf{R}_{\geqslant 0}$.
Output: $\delta'_{\mathcal{T}}$, the estimate of the optimal delta $\delta_{\mathcal{T}}$ with respect to the tuple $(\mathcal{M}, \mathcal{T}, \varepsilon)$.
1) For each pair of neighbors $D \simeq D'$ where $D \in \mathcal{T}$, use algorithm B with n samples compute the estimate of $\delta_{D,D'}$ and the estimate of $\delta_{D',D}$. Denote the maximum among these estimates as $\delta'_{\mathcal{T}}$.
2) Output $\delta'_{\mathcal{T}}$.

Figure 2: $\mathcal{A}^{B}_{\mathcal{C},t}$, an algorithm for estimating the optimal delta with respect to the tuple $(\mathcal{M}, \mathcal{T}, \varepsilon)$

Note on the use of kNN. kNN is a convenient choice for estimating Bayes error in our Eureka framework due to the rich literature on the topic which allows us to prove accuracy guarantees. However kNN is by no means the only estimator that could be used. The literature on kNN and a discussion of the flexibility within the Eureka framework to choose other Bayes error estimators is given in Section 3.2.

DENSITY ASSUMPTION AND DISCRETE OBSERVABLES. Our results as stated require the technical assumption that a mechanism's (random) output possess a density. This is a standard assumption in the ML literature and essentially amounts to the observable being smoothly varying. In fact, common mechanisms that noise their output via a distribution with density (e.g., Laplace, Gaussian), automatically satisfy the above smoothness condition on the density. One can easily generalize this to a discrete observable because a discrete distribution can be approximated arbitrarily closely by a smooth one-dimensional density. This means the Bayes/optimal risk between the two discrete distributions is arbitrarily close to the Bayes risk between the two arbitrarily close smooth approximations.

NOTE ON THE CURSE OF DIMENSIONALITY: The constant c_d in the kNN convergence theorem (Thm.2) is exponential in the dimension of the output, often dubbed "the curse of dimensionality" in the ML literature. In the context of our use of kNN, this implies that our kNN-based privacy estimator requires more samples to maintain the same accuracy as dimension increases. By utilizing a classifier that aims to improve the dependency on dimension, e.g., a neural-network, one can lower the sample requirements, as demonstrated in Section 8.

NOTE ON CHOOSING A DISTANCE FUNCTION: One has the flexibility to choose the distance metric in a kNN classifier. Hence, the kNN-based estimator is very general and can be applied to a wide range of mechanisms. Choosing the best distance metric may not be obvious and may depend on the mechanism. For example, in our use cases, we used the Euclidean distance for mechanisms with real and continuous outputs (like the Laplacian and Gaussian) and the hamming distance for (a variant of) sparse vector technique (SVT) that outputs a bit-vector. Automating (and optimizing) the choice of the distance metric is an interesting avenue for future research.

7. Distributional Differential Privacy

As an extension of our results, we present the *first* privacy estimator for $(\varepsilon, \delta, \Delta)$-*distributional differential privacy* (Def. 4), given Δ contains database distributions where each entry is independently distributed. Of importance, by considering databases as random variables that model a level of adversarial uncertainty about the data, DDP—unlike DP—can formally measure the privacy of even deterministic mechanisms. This means, for the first time, we have shown a method to heuristically estimate the privacy of deterministic mechanisms (under independently distributed data).

First, we observe that DDP under the independence assumption (Def. 4) is very similar to DP. This allows us to define an approximate privacy estimator in a similar manner.

Definition 15. *Let Δ be a set of distributions on size-m databases where each row is independently distributed. We say the privacy parameter $\delta_{\texttt{DDP}}$ is optimal with respect to the tuple $(\mathcal{M}, \Delta, \varepsilon)$ if*

$$\delta_{\texttt{DDP}} = \max\Big(\max_{\pi\in\Delta, i\in[m], x, x'\in\mathcal{U}, \mathcal{S}\subseteq\mathcal{O}} \Pr_{D\sim\pi}[\mathcal{M}(D)\in\mathcal{S}|D_i = x] - e^{\varepsilon}\Pr_{D\sim\pi}[\mathcal{M}(D)\in\mathcal{S}|D_i = x'],\ \ 0\Big).$$

We say $\delta'_{\texttt{DDP}}$ is a α-tight bound with respect to $(\mathcal{M}, \Delta, \varepsilon)$, if

$$|\delta'_{\texttt{DDP}} - \delta_{\texttt{DDP}}| \leqslant \alpha.$$

Definition 16 (Approximate DDP Estimator). *An algorithm is a (α, β)-Approximate DDP Estimator for $\mathcal{C}$ if for every $\mathcal{M} \in \mathcal{C}$, a set of distributions Δ and $\varepsilon \in \mathsf{R}_{\geqslant 0}$, with black-box access to $\mathcal{M}$, with probability at least $1-\beta$, it provides α-tight bound with respect to the tuple $(\mathcal{M}, \Delta, \varepsilon)$, where $\alpha, \beta \in [0, 1)$, and $|\Delta| \leqslant t$ for some $t \in \mathcal{N}^{+}$.*

Our DDP estimator $\mathcal{A}^{B}_{\mathcal{C},\Delta}$, described formally in Fig. 3, is essentially the same as our relative DP estimator, except it is even simpler—here, we only need to run our estimator on the distributions in Δ, rather than enumerating all databases in $\mathcal{T}$. The accuracy of $\mathcal{A}^{B}_{\mathcal{C},\Delta}$ is thus a corollary that can be derived similarly as that of Corollary 3 if we instantiate $\mathcal{A}^{B}_{\mathcal{C},\Delta}$ based on the kNN classifier.

Corollary 4. *Consider the set of mechanisms $\mathcal{C} = \mathcal{U}^m \mapsto \mathsf{R}^d$ whose output distribution has a density. Let the algorithm B be $\mathcal{A}^{\texttt{kNN}}_{\mathcal{C}}$ with n samples, shown in Fig. 2. The algorithm $\mathcal{A}^{B}_{\mathcal{C},\Delta}$, shown in Figure 3, is a (α, β)-Approximate DDP Estimator for $\mathcal{C}$, where $\alpha = 24e^{\varepsilon}c_d\sqrt{\ln(4mt|\mathcal{U}|^2/\beta)/n} + 2e^{\varepsilon}\sqrt{\ln(4mt|\mathcal{U}|^2/\beta)/n}$, $\beta \in (0, 1)$.* [13]

13. Recall that $\mathcal{U}$ is the space of values each entry in the database can take (see Def. 1).

Input: A binary classification algorithm B with n samples, mechanism $\mathcal{M} \in \mathcal{C}$, privacy parameter $\varepsilon \in \mathsf{R}_{\geqslant 0}$, and set of distributions Δ.
Output: $\delta'_{\texttt{DDP}}$, the estimate of the optimal delta $\delta_{\texttt{DDP}}$ with respect to the tuple $(\mathcal{M}, \Delta, \varepsilon)$.

Let $X_{x,i,\pi}$ denote the random variable outputting by the following experiment: sample a database D according to distribution π. Set the i-th row of D to records x. Return $\mathcal{M}(D)$.

Let $[X_{x,i,\pi}]_{\varepsilon}$ denote the random variable obtained by tossing a biased coin c where $\Pr[c = 1] = e^{-\varepsilon}$, and receiving value $X_{x,i,\pi}$ if $c = 1$ or receiving value $\perp$ (a null value not in the range of $\mathcal{M}$) otherwise.

Let $\mathcal{P}$ denote the distribution of a random variable, which is obtained by tossing a fair coin b, and receiving tuple $(X_{x',i,\pi}, 1)$ if $b = 1$ or receiving value $([X_{x,i,\pi}]_{\varepsilon}, 0)$ otherwise.

1) Initialize $n_1 \leftarrow n/2$, $n_2 \leftarrow n/2$, and $\delta'_{\texttt{DDP}} \leftarrow 0$.
2) For all $\pi \in \Delta, i \in [m], x, x' \in \mathcal{U}$
   a) Initialize $r \leftarrow 0$.
   b) Sample n_1 training points $(o_1, b_1), \cdots, (o_{n_1}, b_{n_1})$ according to joint distribution $\mathcal{P}$.
   c) Taking the n_1 training points as inputs, classification algorithm B outputs a classifier $h^{\mathsf{B}}_{n_1}$.
   d) Repeat the process n_2 times: ▷ Estimate risk function of classifier $h^{\mathsf{B}}_{n_1}$ with n_2 testing samples.
      i) Sample a testing point (o, b) according to joint distribution $\mathcal{P}$.
      ii) Predict the sample o's label using the trained classifier: $b' = h^{\mathsf{B}}_{n_1}(o)$. If $b' \neq b$, $r \leftarrow r + 1/n_2$.
   e) Update $\delta'_{\texttt{DDP}} \leftarrow max\left(\delta'_{\texttt{DDP}}, 1 - 2e^{\varepsilon} r\right)$.
3) Output $\delta'_{D,D'}$.

Figure 3: $\mathcal{A}^{B}_{\mathcal{C},\Delta}$, an algorithm for estimating the optimal delta $\delta_{\texttt{DDP}}$ with respect to the tuple $(\mathcal{M}, \Delta, \varepsilon)$

8. Validation and Benchmarking

We demonstrate the applicability and accuracy of our theoretical framework in proof-of-concept implementations of our estimator on a Dell compute node with two 64-core AMD Epyc 7662 "Rome" processors and 256 GB memory. We instantiate our estimator with two different classifiers: kNN and a neural network. These implementations (1) validate our theory and benchmark/stress-test our estimator, (2) showcase our estimator in application scenarios, and (3) demonstrate using our estimator for DDP.

8.1. Benchmarking and Validating our Theory

Our first two sets of experiments (see Figures 4a and 4b) compute the privacy spectrum of the simple Laplace and Gaussian mechanisms, denoted as $\mathcal{M}_{L,\varepsilon}$ and $\mathcal{M}_{G,\varepsilon,\delta}$ respectively (We recall these mechanisms in Definitions 17 and 18 in appendix B.) The figures show that our empirically estimated spectrum is a near exact match with the analytically computed optimal δ for these mechanisms (see Lemma 2 and Lemma 3). It is worth noting that to our knowledge previous theoretical δ given (by well-known bounds [29]) for Gaussian is loose[14], unlike our Lem. 3.

We further note that (besides analytically computing the above privacy spectrum curves) the experiments do not prove something new about the Laplacian and Gaussian mechanism themselves; but they do serve as empirical evidence of our estimator's accuracy, and using such mechanisms with known theoretical behavior is the best and only way to demonstrate that our estimator does deliver on its (theoretically predicted) accuracy. Looking ahead, in Section 8.2.2 we show how our estimator can be used to estimate the spectrum of more complex (and/or heuristic) mechanisms.

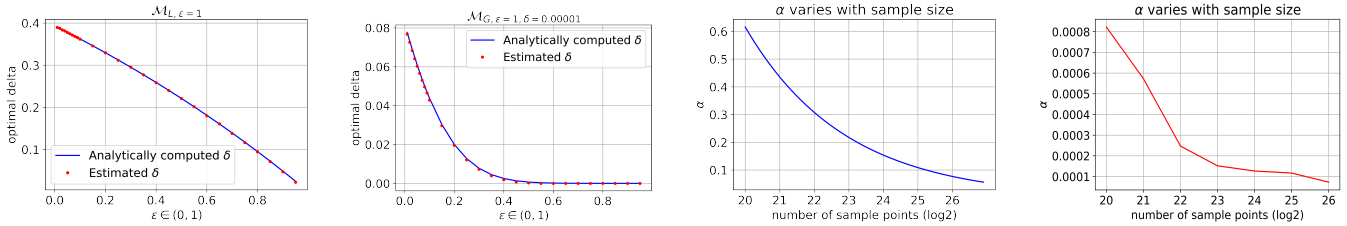
The next set of experiments (Figure 4c and 4d) demonstrates that the accuracy achieved empirically by our (kNN-based) estimator outperforms the theoretical accuracy α of the kNN-based estimator even for a small number of samples. This demonstrates that the asymptotic accuracy of the estimator kicks in already for small size experiments. Such experiments are common in ML theory to validate use of asymptotic behavior to justify practice. (We only show results of the Laplacian mechanism; we have verified the same behavior for the Gaussian mechanism as well.)

Testing discrete and high-dimensional output. Our estimator also applies to mechanisms with discrete[15] or high-dimensional output. First, we test our estimator on the exponential mechanism with the counting query [35], which has discrete output. For demonstration purposes, we use a simple 3-row database $\{0, 1, 1\}$. We set $\varepsilon = 2$. The results (see Figure 5a) show that our estimator empirically produces almost identical privacy spectrum as the analytically computed one for the above mechanism.

Second, we extend our initial two experiments (low-dimensional Laplacian and Gaussian) to high-dimension, to estimate the privacy spectrum of the Laplacian and the Gaussian when the input and output are vectors. Here, the kNN classifier becomes inefficient due to the exponential dependence of theoretically required sample size on dimension. Therefore, we used a neural-network classifier. The results (see Figures 5b and 5c) demonstrate that even in

14. Since Gaussian standard deviation is $\sqrt{\mathcal{N}(0, \frac{2\log(1.25/\delta)}{\varepsilon^2})}$, the δ as the function of ε (the top green curve in Figure 6a,) is very loose.

15. See also the discussion on kNN on Page 11.



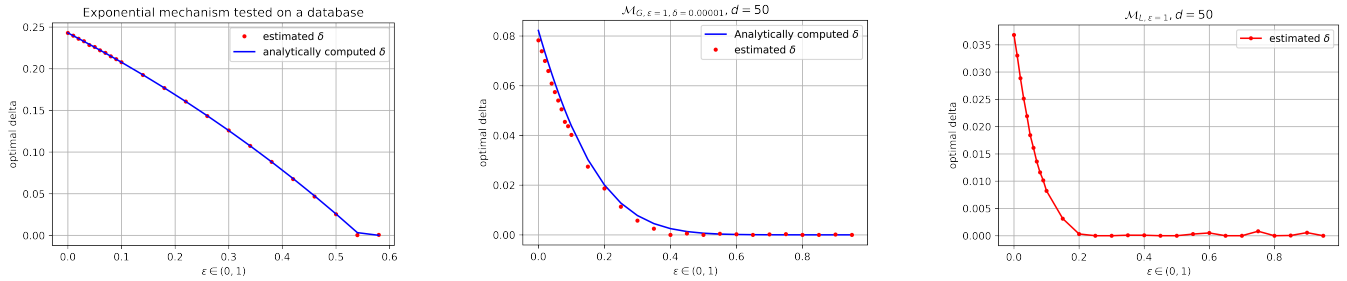
(a) Analytical computed optimal δ compared with estimated δ for $\mathcal{M}_{L,\varepsilon}$

(b) Analytical computed optimal δ compared with estimated δ for $\mathcal{M}_{G,\varepsilon,\delta}$

(c) Theoretical accuracy α of estimated δ vs. number of samples (Theorem 2).

(d) Empirical accuracy α of estimated δ vs. number of samples (for $\mathcal{M}_{L,\varepsilon}$).

Figure 4: Accuracy check for our DP estimator using kNN classifier

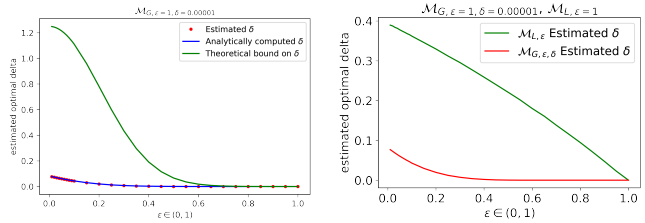


(a) Estimated δ for Exponential mechanism with counting query on a database. Our estimator using kNN can estimate mechanism with discrete outputs.

(b) Analytical computed optimal δ compared with estimated δ for $\mathcal{M}_{G,\varepsilon,\delta}$ with 50-dimensions input

(c) Estimated δ for $\mathcal{M}_{L,\varepsilon}$ with 50-dimensions input

Figure 5: Efficacy check for our DP estimators: Handling High Dimensional and Discrete Outputs



(a) Loose theoretical (top curve) [29, Theorem A.1] vs. actual (analytically computed and estimated) (ε, δ)-privacy spectrum of $\mathcal{M}_{G,\varepsilon,\delta}$ Lemma 3

(b) Estimated (ε, δ)-privacy spectrum of $\mathcal{M}_{G,\varepsilon,\delta}$ and $\mathcal{M}_{L,\varepsilon}$. We see $\mathcal{M}_{G,\varepsilon,\delta}$ (bottom curve) achieves better (smaller) δ.

Figure 6: Application 1: Comparing mechanism privacy

high-dimension, our methodology recovers the analytically computed privacy spectrum.

8.2. Applications of our Estimator

Now we showcase three useful applications of our privacy estimation framework: (1) Comparing the *(differential) privacy spectrum* (i.e., the tradeoff between ε and δ) of two mechanisms, (2) Estimating the privacy of more complex mechanisms, and (3) Verifying mechanism implementations.

8.2.1. Comparing Mechanisms. The (ε, δ) privacy-spectrum generated by our privacy estimator can be used for in-depth comparison of two mechanisms. Let us consider the following example: $\mathcal{M}_{L,\varepsilon}$ and $\mathcal{M}_{G,\varepsilon,\delta}$ are two algorithms whose output is noised so that they give the privacy guarantees of $(\varepsilon, \delta) = (1, 0)$ via Laplace mechanism and $(\varepsilon, \delta) = (1, 0.00001)$ via Gaussian mechanism, respectively. Typically, one would consider $\mathcal{M}_{L,\varepsilon}$ an overall better mechanism. However, as we argue below, this is not always the case, and the (ε, δ) spectrum of these mechanisms leads to a more informed comparison. To demonstrate this, we analytically computed the spectrum for the Laplacian and the Gaussian mechanisms[16] In Figure 6, we compared the analytical bound of the Gaussian to (a) its standard loose bound from [29] and (b) to our new bound on the Laplacian.[17] Our results demonstrate that the superiority of the Laplacian is not absolute—$\mathcal{M}_{G,\varepsilon,\delta}$ provides a stronger (smaller δ) DP guarantee for most settings of ε, something which is not immediate from the much looser bounds from [29]. The above demonstrates that deeper insight on the privacy of a mechanism we can obtain by looking at its privacy spectrum (rather than a single (ε, δ) pair).

Importantly, since we already demonstrated that our estimator's output matches the analytically computed spectrum (Figure 4b and 4a), one can use it to arrive at the same conclusions empirically without the need to analytically compute the mechaninsms' privacy specreum which can be

16. In fact, for the Gaussian mechanism, [36] already includes an alternative way to analytically compute it.

17. For the Laplacian, existing analytic computation do not incorporate δ and only apply to ε-DP.

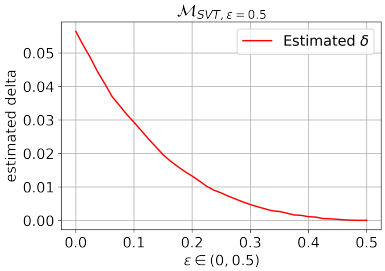


Figure 7: Estimated (ε, δ)-spectrum of $\mathcal{M}_{\mathbf{SVT},\varepsilon}$ parameterized with $\varepsilon = 0.5, \delta = 0$ (same parameters as in [11]).

hectic, or even infeasible for more complicated mechanisms, such as the SVT discussed below, or more complicated ML tasks. This reinforces our belief that an estimator like Eureka with theory-backed accuracy can be an important tool in the hands of domain-experts.

8.2.2. Empirically Estimating Privacy of Complex Mechanisms. A major application of our method is for estimating the privacy of more heuristic approaches. Offering further evidence of our estimator's quality and applicability, we compute the DP spectrum computed for SVT for which analytical tight privacy bounds are not known. Our experiment (see Figure 7) demonstrate that the privacy (spectrum) computed by our estimator is similar to the state of the art computation by [11]. We view applying our estimator to privatizing more complex mechanisms, e.g., randomized machine learning algorithms, as a very promising research direction, albeit beyond the scope of this work which aims at introducing, analyzing, and validating the theory of our framework, showing the tractability of our estimator, and demonstrating its competitiveness for common privacy estimator applications.

8.2.3. Verifying Mechanism Implementation. A common use of privacy estimators has been in verifying (claims about) the privacy of DP mechanisms (e.g., [10], [8]). One common benchmark for this task is to detect buggy implementations of SVT. We compare our implementation with the state-of-the-art tailored for the task, by comparing the estimated privacy of the correct mechanism $\mathcal{M}_{\mathbf{SVT},\varepsilon}$ (discussed above) versus the buggy ones $\mathcal{M}_{\mathbf{SVT2},\varepsilon}$ [20, Alg. 4] and $\mathcal{M}_{\mathbf{SVT3},\varepsilon}$ [20, Alg. 5][18]. Fig. 9a shows that we can detect the worse (higher δ) privacy of the buggy implementations, with estimated δ comparable to [10]. We further used our estimator on the buggy-implementation benchmarks for noisy histogram and noisy max, proposed in [8]. The results in Fig. 9b and Fig. 9c show that our estimator can detect the bug as effectively as the state of the art [10]. We have also tested buggy versions of Laplacian and Gaussian that lead to the same conclusion, that our estimator can detect bugs there too; due to space limitations, the relevant figures are moved to the full version. We believe that these experiments provide credible evidence of our claim that our estimator is appropriate for non-experts

18. In our experiments, we use $k = 40$ queries, and for simplicity consider integer-output queries and thresholds that are no more than 2 away from the true query output

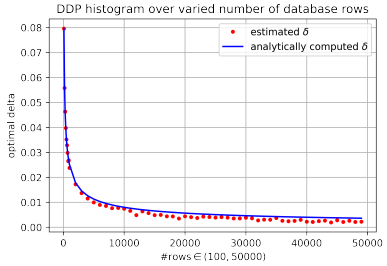


Figure 8: Estimated (ε, δ)-spectrum of DDP histogram [33].

to deploy in complex applications while producing quality privacy estimates.

8.3. Applying to DDP

We demonstrate the applicability of our methods to DDP by presenting the first estimator for the (noiseless) histogram mechanism. Using the setting from [33] (databases with uniform, i.i.d. rows with records in $\{0, 1\}$ and $\varepsilon = 0$), our estimates (Fig. 8) follows closely with the theoretical $\delta = O(1/\sqrt{n})$ curve [33] (which can be made concrete in the simple case of each row being $\in \{0, 1\}$) between δ and the number of database rows. We note that this last benchmark only scratches the surface of what can be obtained by applying our estimator to the design of DDP mechanisms which we consider an excellent question for future research. Our goal here was to introduce the theoretical framework and validate it in a wide range of use cases.

9. Conclusion/Future Work

We presented a methodology for black-box privacy estimators that combines DP with ML (classifiers) and allows plug-and-play use of different classifiers to achieve desirable guarantees. We introduced the privacy spectrum as a more intuitive way to quantify the privacy guarantee of a given mechanism and devised, using our methodology, the first general estimator for the ε, δ privacy spectrum with tight theoretical accuracy bounds. We proved an impossibility result of (ε, δ)-DP estimators when the input space is unlimited, then circumvented it with *relative DP*, an intuitive relaxation that formalizes a limitation on input space. We believe this limitation can be of independent interest for application of DP in complex machine learning algorithms, especially when the available datasets are limited, e.g., genomic or medical research. Building on our first step in applying our methods to estimate DDP, an interesting open problem is to take this further to other non-standard DP definitions. Lastly, we show that a kNN-based version of our estimator achieves high experimental and theoretical accuracy testing mechanisms with low dimensional output, whereas the neural network-based estimator preserves high efficiency even on high dimensions. This demonstrates promise in alternative instantiations that selectively plug in different classifiers to our estimator based on desired properties. Our experiments provide first proof of concept implementations and highlight computational bottlenecks, in particular in relation to the

size of δ, of the current state of affairs. Improving on this bottleneck via ML, algorithmic, and optimized engineering techniques in an interesting future direction.

Acknowledgment

We would like to thank Maksim Tsikhanovich for initial fruitful discussions on linking measures of privacy to Bayes optimal classification problems. In addition, we thank him for pointing us to his code at [37] which was used in our experiments to implement the empirical bootstrapping method. This piece of code helps us understand the relationship between our estimator's empirical tightness and the number of samples.

References

[1] U.S. Census Bureau, *2020 Census Disclosure Avoidance Handbook*, March 2020. [Online]. Available: https://www2.census.gov/library/publications/decennial/2020/2020-census-disclosure-avoidance-handbook.pdf

[2] Y. Lu, M. Magdon-Ismail, Y. Wei, and V. Zikas, "Privacy-utility tradeoff of OLS with random projections," *CoRR*, vol. abs/2309.01243, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2309.01243

[3] B. Balle, G. Barthe, and M. Gaboardi, "Privacy amplification by subsampling: Tight analyses via couplings and divergences," in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., 2018, pp. 6280–6290. [Online]. Available: https://proceedings.neurips.cc/paper/2018/hash/3b5020bb891119b9f5130f1fea9bd773-Abstract.html

[4] D. Zhang and D. Kifer, "Lightdp: Towards automating differential privacy proofs," in *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages*, 2017, pp. 888–901.

[5] Y. Wang, Z. Ding, G. Wang, D. Kifer, and D. Zhang, "Proving differential privacy with shadow execution," in *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*, 2019, pp. 655–669.

[6] Y. Wang, Z. Ding, D. Kifer, and D. Zhang, "CheckDP: An automated and integrated approach for proving differential privacy or finding precise counterexamples," in *ACM CCS 2020*, J. Ligatti, X. Ou, J. Katz, and G. Vigna, Eds. ACM Press, Nov. 2020, pp. 919–938.

[7] H. Zhang, E. Roth, A. Haeberlen, B. C. Pierce, and A. Roth, "Testing differential privacy with dual interpreters," *Proc. ACM Program. Lang.*, vol. 4, no. OOPSLA, nov 2020.

[8] Z. Ding, Y. Wang, G. Wang, D. Zhang, and D. Kifer, "Detecting violations of differential privacy," in *ACM CCS 2018*, D. Lie, M. Mannan, M. Backes, and X. Wang, Eds. ACM Press, Oct. 2018, pp. 475–489.

[9] B. Bichsel, T. Gehr, D. Drachsler-Cohen, P. Tsankov, and M. T. Vechev, "DP-finder: Finding differential privacy violations by sampling and optimization," in *ACM CCS 2018*, D. Lie, M. Mannan, M. Backes, and X. Wang, Eds. ACM Press, Oct. 2018, pp. 508–524.

[10] B. Bichsel, S. Steffen, I. Bogunovic, and M. T. Vechev, "DP-sniper: Black-box discovery of differential privacy violations using classifiers," in *2021 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 2021, pp. 391–409.

[11] X. Liu and S. Oh, "Minimax optimal estimation of approximate differential privacy on neighboring databases," *Advances in neural information processing systems*, vol. 32, 2019.

[12] I. Mironov, "Rényi differential privacy," in *2017 IEEE 30th computer security foundations symposium (CSF)*. IEEE, 2017, pp. 263–275.

[13] C. Dwork and G. N. Rothblum, "Concentrated differential privacy," *arXiv preprint arXiv:1603.01887*, 2016.

[14] M. Bun and T. Steinke, "Concentrated differential privacy: Simplifications, extensions, and lower bounds," in *Theory of Cryptography Conference*. Springer, 2016, pp. 635–658.

[15] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, ser. Stochastic Modelling and Applied Probability. Springer, 1996, vol. 31.

[16] R. Bassily, A. Groce, J. Katz, and A. Smith, "Coupled-worlds privacy: Exploiting adversarial uncertainty in statistical data privacy," in *54th FOCS*. IEEE Computer Society Press, Oct. 2013, pp. 439–448.

[17] A. Antos, L. Devroye, and L. Gyorfi, "Lower bounds for bayes error estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 7, pp. 643–645, 1999.

[18] Ö. Askin, T. Kutta, and H. Dette, "Statistical quantification of differential privacy: A local approach," in *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2022, pp. 402–421.

[19] J. Zhang, X. Xiao, and X. Xie, "Privtree: A differentially private algorithm for hierarchical decompositions," in *SIGMOD Conference*. ACM, 2016, pp. 155–170.

[20] M. Lyu, D. Su, and N. Li, "Understanding the sparse vector technique for differential privacy," *Proc. VLDB Endow.*, vol. 10, no. 6, pp. 637–648, 2017.

[21] Ú. Erlingsson, V. Pihur, and A. Korolova, "RAPPOR: randomized aggregatable privacy-preserving ordinal response," in *CCS*. ACM, 2014, pp. 1054–1067.

[22] T. Humphries, M. Rafuse, L. Tulloch, S. Oya, I. Goldberg, U. Hengartner, and F. Kerschbaum, "Differentially private learning does not bound membership inference," *arXiv preprint arXiv:2010.12112*, 2020.

[23] Ú. Erlingsson, I. Mironov, A. Raghunathan, and S. Song, "That which we call private," *arXiv preprint arXiv:1908.03566*, 2019.

[24] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *2018 IEEE 31st computer security foundations symposium (CSF)*. IEEE, 2018, pp. 268–282.

[25] K. Chatzikokolakis, G. Cherubin, C. Palamidessi, and C. Troncoso, "The bayes security measure," *arXiv preprint arXiv:2011.03396*, 2020.

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

[27] A. C. Gilbert and A. McMillan, "Property testing for differential privacy," in *Allerton*. IEEE, 2018, pp. 249–258.

[28] C. Dwork, "Differential privacy (invited paper)," in *ICALP 2006, Part II*, ser. LNCS, M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds., vol. 4052. Springer, Heidelberg, Jul. 2006, pp. 1–12.

[29] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3–4, p. 211–407, Aug. 2014. [Online]. Available: https://doi.org/10.1561/0400000042

[30] R. Bhaskar, A. Bhowmick, V. Goyal, S. Laxman, and A. Thakurta, "Noiseless database privacy," in *ASIACRYPT 2011*, ser. LNCS, D. H. Lee and X. Wang, Eds., vol. 7073. Springer, Heidelberg, Dec. 2011, pp. 215–232.

[31] S. Leung and E. Lui, "Bayesian mechanism design with efficiency, privacy, and approximate truthfulness," in *International Workshop on Internet and Network Economics*. Springer, 2012, pp. 58–71.

[32] Y. Duan, "Privacy without noise," in *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, ser. CIKM '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 1517–1520. [Online]. Available: https://doi.org/10.1145/1645953.1646160

[33] L. Ao, Y. Lu, L. Xia, and V. Zikas, "How private are commonly-used voting rules?" in *Conference on Uncertainty in Artificial Intelligence*. PMLR, 2020, pp. 629–638.

[34] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 13–30, 1963.

[35] N. Li, M. Lyu, D. Su, and W. Yang, *Differential Privacy: From Theory to Practice*, ser. Synthesis Lectures on Information Security, Privacy, & Trust. Morgan & Claypool Publishers, 2016. [Online]. Available: https://doi.org/10.2200/S00735ED1V01Y201609SPT018

[36] B. Balle and Y. Wang, "Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, ser. Proceedings of Machine Learning Research, J. G. Dy and A. Krause, Eds., vol. 80. PMLR, 2018, pp. 403–412. [Online]. Available: http://proceedings.mlr.press/v80/balle18a.html

[37] M. Tsikhanovich, "empirical privacy," https://github.com/maksimt/empirical_privacy, 2019.

Appendix

1. Additional Figures for Section 8

We include some of the figures referenced in Section 8: Figures 9a, 9b, and 9c show that our estimator can be used to distinguish buggy implementations of SVT, noisy histogram, and noisy max mechanisms, respectively.

2. Proofs

Proof of Theorem 3. We will prove the theorem by

1) constructing two mechanisms $\mathcal{M}$ and $\mathcal{M}_D$, where $\mathcal{M}_D$ is a mechanism parameterized with a database D.
2) showing that there does not exist a polynomial time algorithm P that can distinguish between $\mathcal{M}$ and $\mathcal{M}_D$ if D is randomly chosen.
3) proving by contradiction that if the algorithm E_ε defined in the lemma exists, then we can turn it into a distinguisher P (which was proven impossible).

We start by constructing two mechanisms $\mathcal{M}$ and $\mathcal{M}_D$. Let $\mathcal{M} : \{0,1\}^n \mapsto \{0,1\}$ and $\mathcal{M}_D : \{0,1\}^n \mapsto \{0,1\}$ be two randomized mechanisms. Let $D \in \{0,1\}^n$. We define $\mathcal{M}$ as the following: no matter what input in the domain it takes, $\mathcal{M}$ outputs 0 with probability $\frac{1}{2}$ otherwise outputs 1 with probability $\frac{1}{2}$. We define $\mathcal{M}_D$ as the following: given any input x not equal to D it outputs $\mathcal{M}(x)$ otherwise $\mathcal{M}_D$ outputs 0 with probability 0 and 1 with probability 1.

We know that $\mathcal{M}$ is $(0,0)$-differential private, because its output is independent of its input. Also, we know that $\mathcal{M}_D$ is $(0,1)$-differential private, because its output is deterministic when given D.

Then, we define the following game for algorithm P: Choose database D uniformly at random from $\{0,1\}^n$. Toss a fair coin b, and give the algorithm P black-box access to either $\mathcal{M}$ or $\mathcal{M}_D$ based on b. The algorithm P wins if it can correctly decide whether it was given $\mathcal{M}$ or $\mathcal{M}_D$.

Since P is running in polynomial time, and has only black-box access to the mechanism, this means we can consider P's output as a randomized function of its $\mathrm{poly}(n)$ queries $D_1, D_2, \cdots$ (made possibly adaptively) to the mechanism. Since $\mathcal{M}$'s and $\mathcal{M}_D$'s outputs only differ on input D, and D is chosen uniformly at random, it means the probability that P queries D is negligible in n. In other words, P can only win with at best negligibly better probability than guessing $(1/2)$.

We now prove by contradiction that E_ε defined in the lemma does not exist. Suppose for contradiction that E_ε does indeed exist. Then, let P do the following: given a mechanism (one of $\mathcal{M}$ or $\mathcal{M}_D$), feed this mechanism and $\varepsilon = 0$ to E_ε. If E_ε says an estimate $\delta' \leqslant \alpha$, P guesses that it was given $\mathcal{M}$. Else, it guesses that it was given $\mathcal{M}_D$. Since, with probability $\frac{1}{2} + \nu(n)$, E_ε should always give some estimate $\delta' \in [0, \alpha]$ given $\mathcal{M}$, and some estimate $\delta' \in [1-\alpha, 1]$ given $\mathcal{M}_D$, it means P should be correct with probability at least $\frac{1}{2}+\nu(n)$. This contradicts the conclusion of (2), meaning E_ε does not exist. □

Proof of Prop. 1, relative DP-to-DP. This proposition holds by definition of (relative) differential privacy. □

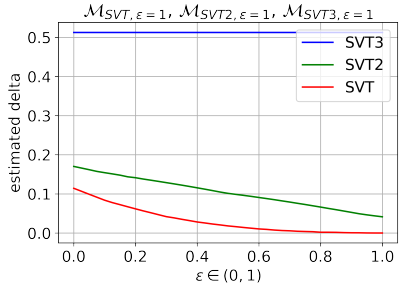
Proof of Prop. 2, $\mathcal{T}$-scalability. By the relative DP definition and the proposition's condition, the mechanism $\mathcal{M}$ satisfies that, for every pair of neighboring databases $D \simeq D'$ and $D' \simeq D : D \in \mathcal{T}$ and subset $\mathcal{S} \subseteq \mathtt{Range}(\mathcal{M})$,

$$\begin{aligned}\Pr[\mathcal{M}(D) \in \mathcal{S}] &\leqslant e^{\varepsilon_i} \Pr[\mathcal{M}(D') \in \mathcal{S}] + \delta_i \\ &\leqslant e^{\max_{i\in[k]} \varepsilon_i} \Pr[\mathcal{M}(D') \in \mathcal{S}] + \max_{i\in[k]} \delta_i,\end{aligned}$$

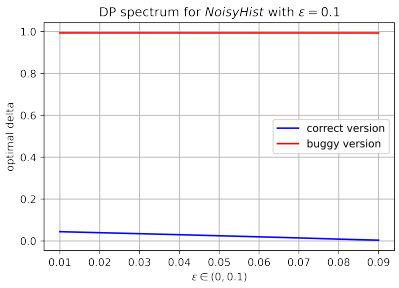
which completes the proof. □

Proof of Prop. 3, Parallel composition. Let $D = (D_1, \cdots, D_k)$ be a arbitrary database from the set $\mathcal{T}_1 \times \cdots \times \mathcal{T}_k$. Let $D' = (D'_1, \cdots, D'_k)$ be a arbitrary neighbor of D. Without loss of generality, let D_j, $j \in [k]$, to be the database such that $D_j \neq D'_j$, and we have $D_i = D'_i$ for $i \in [k]$ and $i \neq j$. For every subset $\mathcal{S} \subseteq \mathtt{Range}(\mathcal{M})$, we have

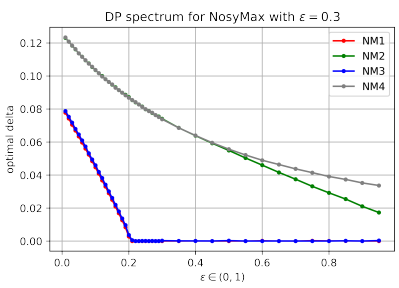
$$\begin{aligned}&\Pr[\mathcal{M}(D) \in \mathcal{S}] \\ &= \Pr[(\mathcal{M}_1(D_1), \cdots, \mathcal{M}_k(D_k)) \in (\mathcal{S}_1, \cdots, \mathcal{S}_k)] \\ &= \prod_{i\in[k]} \Pr[\mathcal{M}_i(D_i) \in \mathcal{S}_i] \\ &= \Pr[\mathcal{M}_j(D_j) \in \mathcal{S}_j] \prod_{i\in[k]\setminus\{j\}} \Pr[\mathcal{M}_i(D_i) \in \mathcal{S}_i] \\ &\leqslant (e^{\varepsilon_j} \Pr[\mathcal{M}_j(D'_j) \in \mathcal{S}_j] + \delta_j) \prod_{i\in[k]\setminus\{j\}} \Pr[\mathcal{M}_i(D'_i) \in \mathcal{S}_i] \\ &\leqslant e^{\varepsilon_j} \Pr[\mathcal{M}_j(D'_j) \in \mathcal{S}_j] \prod_{i\in[k]\setminus\{j\}} \Pr[\mathcal{M}_i(D'_i) \in \mathcal{S}_i] + \delta_j \\ &= e^{\varepsilon_j} \Pr[\mathcal{M}(D') \in \mathcal{S}] + \delta_j \\ &\leqslant (\max_{i\in[k]} e^{\varepsilon_i}) \Pr[\mathcal{M}(D') \in \mathcal{S}] + (\max_{i\in[k]} \delta_i),\end{aligned}$$



(a) Estimated (ε, δ) spectrum of $\mathcal{M}_{\mathbf{SVT},\varepsilon}$ [20, Alg.1] and its two variants. $\mathcal{M}_{\mathbf{SVT2},\varepsilon}$ [20, Alg.4] and $\mathcal{M}_{\mathbf{SVT3},\varepsilon}$ [20, Alg.5] are not $(\varepsilon = 1, \delta = 0)$-DP.



(b) Estimated (ε, δ) spectrum of Noisy histogram [8, Alg.9] and its buggy variant. The buggy version [8, Alg.10] is not $(\varepsilon = 0.1, \delta = 0)$-DP.



(c) Estimated (ε, δ) spectrum of four variants of Noisy max. NM1 and NM3 [8, Alg.5, 6] satisfy $(\varepsilon = 0.3, \delta = 0)$-DP while NM2 and NM4 [8, Alg.7, 8] do not.

Figure 9: DP-spectrum of variants SVT, Noisy histogram and Noisy max

which completes the proof. □

Proof of Prop. 4, Sequential composition. Let $D \in \mathcal{T}$ be any pair of neighbors, $D' \simeq D$. For every subset $\mathcal{S} \subseteq \texttt{Range}(\mathcal{M})$, we have

$$\begin{aligned}
&\Pr[\mathcal{M}(D) \in \mathcal{S}] \\
&= \Pr[(\mathcal{M}_1(D), \cdots, \mathcal{M}_k(D)) \in (\mathcal{S}_1, \cdots, \mathcal{S}_k)] \\
&= \prod_{i\in[k]} \Pr[\mathcal{M}_i(D) \in \mathcal{S}_i] \\
&= \prod_{i\in[k-1]} \Pr[\mathcal{M}_i(D) \in \mathcal{S}_i]) \Pr[\mathcal{M}_k(D) \in \mathcal{S}_k] \\
&\leqslant \prod_{i\in[k-1]} \Pr[\mathcal{M}_i(D) \in \mathcal{S}_i] \left(e^{\varepsilon_k} \Pr[\mathcal{M}_k(D') \in \mathcal{S}_k] + \delta_k\right) \\
&\leqslant e^{\varepsilon_k} \Big(\prod_{i\in[k-1]} \Pr[\mathcal{M}_i(D) \in \mathcal{S}_i]) \Pr[\mathcal{M}_k(D') \in \mathcal{S}_k] + \delta_k \\
&\leqslant e^{\sum_{i\in[k]} \varepsilon_i} \Pr[\mathcal{M}(D') \in \mathcal{S}] + \sum_{i\in[k]} \delta_i,
\end{aligned}$$

which completes the proof. □

Proof of Prop. 5. Let $D \in \mathcal{T}$ be any pair of neighbors, $D' \simeq D$. For every subset $\mathcal{S} \subseteq \texttt{Range}(\mathcal{M})$, define set $T = \{t \in \texttt{Range}(\mathcal{M}_1) : f(t) \in \mathcal{S}\}$. We have

$$\begin{aligned}
\Pr[\mathcal{M}(D) \in \mathcal{S}] &= \Pr[f(\mathcal{M}_1(D)) \in \mathcal{S}] \\
&= \sum_{t\in T} \Pr[\mathcal{M}_1(D) = t] \\
&= \Pr[\mathcal{M}_1(D) \in T] \\
&\leqslant e^{\varepsilon} \Pr[\mathcal{M}_1(D') \in T] + \delta, \\
&= e^{\varepsilon} \Pr[\mathcal{M}(D') \in \mathcal{S}] + \delta.
\end{aligned}$$

which completes the proof. □

Proof of Theorem 4. Let $\Delta\left(\left[\mathcal{M}(D)\right]_{\varepsilon}, \mathcal{M}(D')\right)$ be the statistical distance between $\left[\mathcal{M}(D)\right]_{\varepsilon}$ and $\mathcal{M}(D')$. Our plan of proof is the following. We first show the equivalence between the optimal $\delta_{D,D'}$ and the statistical distance $\Delta\left(\left[\mathcal{M}(D)\right]_{\varepsilon}, \mathcal{M}(D')\right)$.

Claim 1. *The following equation between the optimal $\delta_{D,D'}$ with respect to the tuple $(\mathcal{M}, D, D', \varepsilon)$ and the statistical distance $\Delta\left(\left[\mathcal{M}(D)\right]_{\varepsilon}, \mathcal{M}(D')\right)$ holds:*

$$\delta_{D,D'} = \max\left(e^{\varepsilon}\left(\Delta\left(\left[\mathcal{M}(D)\right]_{\varepsilon}, \mathcal{M}(D')\right) - (1 - e^{-\varepsilon})\right), 0\right).$$

Proof of Claim 1 By definition of optimal $\delta_{D,D'}$ in Definition 5, we have

$$\begin{aligned}
\delta_{D,D'} &= \max\left(\max_{\mathcal{S}\subseteq\mathcal{O}} \Pr[\mathcal{M}(D) \in \mathcal{S}] - e^{\varepsilon} \Pr[\mathcal{M}(D') \in \mathcal{S}], 0\right) \\
&= \max\left(e^{\varepsilon} \max_{\mathcal{S}\subseteq\mathcal{O}} \left(e^{-\varepsilon} \Pr[\mathcal{M}(D) \in S] - \Pr[\mathcal{M}(D') \in S]\right), 0\right).
\end{aligned} \tag{1}$$

We first check that the distribution $\left[\mathcal{M}(D)\right]_{\varepsilon}$ has the following property, for all $\mathcal{S} \in \mathcal{O}$ (support of mechanism $\mathcal{M}$),

$$\Pr\left[\left[\mathcal{M}(D)\right]_{\varepsilon} \in \mathcal{S}\right] = e^{-\varepsilon} \Pr[\mathcal{M}(D) \in \mathcal{S}].$$

This is because, for all $\mathcal{S} \in \mathcal{O}$,

$$\begin{aligned}
\Pr[\left[\mathcal{M}(D)\right]_{\varepsilon} \in \mathcal{S}] &= \Pr[c = 1 \wedge \mathcal{M}(D) \in \mathcal{S}] \\
&= \Pr[c = 1] \Pr[\mathcal{M}(D) \in \mathcal{S}] \\
&\qquad (c \text{ and } \mathcal{M}(D) \text{ are independent}) \\
&= e^{-\varepsilon} \Pr[\mathcal{M}(D) \in \mathcal{S}].
\end{aligned}$$

We are given a method to find the statistical distance between two distributions by sampling them. The statistical distance between distributions $\left[\mathcal{M}(D)\right]_{\varepsilon}$ and $\mathcal{M}(D')$ is defined as follows:

$$\Delta\Big(\left[\mathcal{M}(D)\right]_{\varepsilon}, \mathcal{M}(D')\Big) \equiv \max_{\mathcal{S}\subseteq\mathcal{O}} \Big(\Pr\left[\left[\mathcal{M}(D)\right]_{\varepsilon} \in \mathcal{S}\right] - \Pr[\mathcal{M}(D') \in \mathcal{S}]\Big).$$

By construction, $\left[\mathcal{M}(D)\right]_{\varepsilon}$ outputs $\perp$ with probability $1 - e^{-\varepsilon}$, whereas $\mathcal{M}(D')$ outputs $\perp$ with probability zero. Thus, $\perp$ can always be included in the set that maximizes the statistical distance.

$$\begin{aligned}
&\Delta(\left[\mathcal{M}(D)\right]_{\varepsilon}, \mathcal{M}(D')) \\
&= \max_{\mathcal{S}\in\mathcal{O}} \Big(\Pr[\left[\mathcal{M}(D)\right]_{\varepsilon} \in \mathcal{S}] - \Pr[\mathcal{M}(D') \in \mathcal{S}]\Big) \\
&+ \Big((\Pr[\left[\mathcal{M}(D)\right]_{\varepsilon} = \perp] - \Pr[\mathcal{M}(D') = \perp]\Big) \\
&= \max_{\mathcal{S}\in\mathcal{O}} \Big(e^{-\varepsilon} \Pr[\mathcal{M}(D) \in \mathcal{S}] - \Pr(\mathcal{M}(D') \in \mathcal{S})\Big) + (1 - e^{-\varepsilon})
\end{aligned}$$

Then, plug the above equation into the equation 1, we have

$$\delta_{D,D'} = \max(e^{\varepsilon}\left(\Delta\left(\left[\mathcal{M}(D)\right]_{\varepsilon}, \mathcal{M}(D')\right) - (1 - e^{-\varepsilon})\right), 0),$$

which completes the proof.

Secondly, we show the equivalence between risk of the the Bayes classifier $R(h^*_{D,D'})$ and the statistical distance $\Delta\left(\left[\mathcal{M}(D)\right]_{\varepsilon}, \mathcal{M}(D')\right)$.

Claim 2.

$$\Delta\left(\left[\mathcal{M}(D)\right]_{\varepsilon}, \mathcal{M}(D')\right) = 2 \cdot \left(\frac{1}{2} - R(h^*_{D,D'})\right).$$

Proof of Claim 2 The statistical distance can be alternatively defined as

$$\Delta\left(\left[\mathcal{M}(D)\right]_{\varepsilon}, \mathcal{M}(D')\right) = \max_h \Big| \Pr_{x\sim\mathcal{M}(D')}[h(x)=1] - \Pr_{x\sim\left[\mathcal{M}(D)\right]_{\varepsilon}}[h(x)=1]\Big|,$$

where h is any classifier for the distribution $\mathcal{P}$. Then,

$$\begin{aligned}
&\Delta\left(\left[\mathcal{M}(D)\right]_{\varepsilon}, \mathcal{M}(D')\right) \\
&= 2\left(\frac{1}{2}\max_h \Big| \Pr_{x\sim\mathcal{M}(D')}[h(x)=1] - \Big(1 - \Pr_{x\sim\left[\mathcal{M}(D)\right]_{\varepsilon}}[h(x)=0]\Big)\Big|\right) \\
&= 2\left(\max_h \left|\frac{1}{2}\Big(\Pr_{x\sim\mathcal{M}(D')}[h(x)=1] + \Pr_{x\sim\left[\mathcal{M}(D)\right]_{\varepsilon}}[h(x)=0]\Big) - \frac{1}{2}\right|\right) \\
&= 2\Big(\max_h \Big| \Pr_{(x,y)\sim\mathcal{P}}[h(x)=1, y=1] + \Pr_{(x,y)\sim\mathcal{P}}[h(x)=0, y=0] - \frac{1}{2}\Big|\Big) \\
&= 2\Big(\max_h \Big| \Pr_{(x,y)\sim\mathcal{P}}[h(x)=y] - \frac{1}{2}\Big|\Big) \\
&= 2\Big(\max_h \Big|1 - \Pr_{(x,y)\sim\mathcal{P}}[h(x)\neq y] - \frac{1}{2}\Big|\Big) \\
&= 2\Big(\max_h \Big|\frac{1}{2} - R(h)\Big|\Big) \\
&= 2\Big(\frac{1}{2} - R(h^*_{D,D'})\Big).
\end{aligned}$$

Show the equivalence between the optimal $\delta_{D,D'}$ and the risk of the the Bayes classifier $R(h^*)$. Combining the Claim 1 and the Claim 2, it is easy to show that

$$\delta_{D,D'} = \max\left(1 - 2e^{\varepsilon}R(h^*_{D,D'}), 0\right),$$

which completes the proof. □

Proof of Lemma 1. For every $(\mathcal{M}, D, D', \varepsilon)$, and its corresponding distribution $\mathcal{P}$, we have the following. Recall the random variable r as computed in Step 4, Figure 1, is the testing risk for classifier $h^{\mathtt{B}}_{n_1}$ with n_2 testing samples. We could show that r is a good approximate of the risk of the Bayes classifier $R(h^*_{D,D'})$.

Claim 3. *With probability at least* $1-\beta$,

$$|r - R(h^*_{D,D'})| \leqslant g(\mathcal{X}, n/2, \beta/2) + \sqrt{\ln(4/\beta)/n}.$$

Proof of Claim 3 Recall $n_1 = n/2$, defined in Step 1, Fig. 1. By the condition in the Lemma, when the sample size parameter n_1 is large enough, we have that, with probability at least $1-\beta/2$,

$$|R(h^{\mathtt{B}}_{n_1}) - R(h^*_{D,D'})| \leqslant g(\mathcal{X}, n_1, \beta/2) = g(\mathcal{X}, n/2, \beta/2).$$

By Theorem 1, plug in $n_2 = n/2$ (defined in Step 1, Fig. 1), with probability at least $1-\beta/2$, we have

$$|r - R(h^{\mathtt{B}}_{n_1})| \leqslant \sqrt{\ln(4/\beta)/n}.$$

Apply union bound and triangular inequality to the above two inequalities, with probability at least $1-\beta$, we have

$$\begin{aligned}
|r - R(h^*)| &\leqslant |r - R(h^{\mathtt{B}}_{n_1})| + |R(h^{\mathtt{B}}_{n_1}) - R(h^*_{D,D'})| \\
&\leqslant g(\mathcal{X}, n/2, \beta/2) + \sqrt{\ln(4/\beta)/n},
\end{aligned}$$

which completes the proof. Using Claim 3, we could show that $\delta'_{D,D'}$ (defined in Step 5, Fig. 1) is a good approximate of $\delta_{D,D'}$ with respect to $(\mathcal{M}, D, D', \varepsilon)$.

Claim 4. *With probability at least* $1-\beta$,

$$\left|\delta'_{D,D'} - \delta_{D,D'}\right| = 2e^{\varepsilon}g(\mathcal{X}, n/2, \beta/2) + 2e^{\varepsilon}\sqrt{\ln(4/\beta)/n}.$$

Proof of Claim 4

$$\begin{aligned}
&\left|\delta'_{D,D'} - \delta_{D,D'}\right| \\
&= \left|\max\left(1 - 2e^{\varepsilon}r, 0\right) - \delta_{D,D'}\right| && \text{(By Fig. 1, Step 5,)} \\
&= \left|\max\left(1 - 2e^{\varepsilon}r, 0\right) - \max\left(1 - 2e^{\varepsilon}R\left(h^*_{D.D'}\right), 0\right)\right| \\
& && \text{(By Theorem 4)} \\
&\leqslant \left|\left(\left(1 - 2e^{\varepsilon}r\right) - \left(1 - 2e^{\varepsilon}R\left(h^*_{D.D'}\right)\right)\right)\right| \\
&\leqslant 2e^{\varepsilon}\left|r - R(h^*_{D.D'})\right| \\
&= 2e^{\varepsilon}g(\mathcal{X}, n/2, \beta/2) + 2e^{\varepsilon}\sqrt{\ln(4/\beta)/n}. && \text{(By Claim 3)}
\end{aligned}$$

By Claim 4, we have that for every tuple $(\mathcal{M}, D, D', \varepsilon)$ the algorithm $\mathcal{A}^{\mathtt{B}}_{\mathcal{C}}$ provides a $\alpha = 2e^{\varepsilon}g(\mathcal{X}, n/2, \beta/2) + 2e^{\varepsilon}\sqrt{\ln(4/\beta)/n}$ tight bound with probability $1-\beta$. Thus concludes the proof that $\mathcal{A}^{\mathtt{B}}_{\mathcal{C}}$ is a (α, β)-Approximate δ-Estimator for a Pair of Databases for $\mathcal{C}$. □

Proof of Theorem 5. Let q be the number of neighboring databases $D \simeq D'$ where $D \in \mathcal{T}$. Let $\{\delta_1, \cdots, \delta_{2q}\}$ be the set of optimal $\delta_{D,D'}$ (and $\delta_{D',D}$) for each neighboring databases, $\{\delta'_1, \cdots, \delta'_{2q}\}$ (computed in Step 1, Fig. 2) be the set of estimate for $\{\delta_1, \cdots, \delta_{2q}\}$. δ'_1 is the estimate of δ_1, etc.

By Lemma 1, we could say that for each $i \in [2q]$, with probability at least $1 - \beta/2q$,

$$|\delta'_i - \delta_i| \leqslant 2e^{\varepsilon}g(\mathcal{X}, n/2, \beta/4q) + 2e^{\varepsilon}\sqrt{\ln(8q/\beta)/n},$$

By a union bound, with probability at least $1-\beta$,

$$\max_{i\in[2q]} |\delta'_i - \delta_i| \leqslant 2e^{\varepsilon}g(\mathcal{X}, n/2, \beta/4q) + 2e^{\varepsilon}\sqrt{\ln(8q/\beta)/n}. \quad (2)$$

Denote the index of $\delta_{\mathcal{T}}$ in set $\{\delta_1, \cdots, \delta_{2q}\}$ as a. That is $\delta_{\mathcal{T}} = \delta_a = \max_{i\in[2q]} \delta_i$. Denote the index of the maximum estimate in set $\{\delta'_1, \cdots, \delta'_{2q}\}$ as b. That is $\delta'_b = \max_{i\in[2q]} \delta'_i$. The algorithm $\mathcal{A}^{B}_{\mathcal{C},t}$ outputs δ'_b as the estimate of $\delta_{\mathcal{T}}$. Then, with probability at least $1-\beta$,

$$\begin{aligned}
|\delta'_b - \delta_{\mathcal{T}}| &= |\delta'_b - \delta_a| \\
&\leqslant \max\left(|\delta'_b - \delta_b|, |\delta'_a - \delta_a|\right) \\
&\leqslant \max_{i\in[2q]} |\delta'_i - \delta_i|.
\end{aligned} \quad (3)$$

We bound the total number of neighboring databases q. Because the size of the databases set $\mathcal{T}$ is smaller than t, there must exist at most tm (where m is the (maximum) number of database rows) neighbours of databases in $\mathcal{T}$. That is,

$$q \leqslant tm. \tag{4}$$

Combining Inequalities 2, 3 and 4, with probability at least $1-\beta$,

$$|\delta_b' - \delta_{\mathcal{T}}| \leqslant 2e^{\varepsilon} g(\mathcal{X}, n/2, \beta/4tm) + 2e^{\varepsilon}\sqrt{\ln(8tm/\beta)/n},$$

which completes the proof. □

Proof of Corollary 2. The algorithm $\mathcal{A}_{\mathcal{C}}^{\text{kNN}}$ with the classification algorithm kNN is a concrete instantiation of $\mathcal{A}_{\mathcal{C}}^{\text{B}}$, shown in Figure 1. To prove that $\mathcal{A}_{\mathcal{C}}^{\text{kNN}}$ is a (α,β)-Approximate δ-Estimator for a Pair of Databases for $\mathcal{C}$, we could directly plug in the convergence results of kNN into Lemma 1 and then complete the proof.

For every tuple $(\mathcal{M}, D, D', \varepsilon)$, where $\mathcal{M} \in \mathcal{C}$, we have two random variables: $\mathcal{M}(D')$ and $\left[\mathcal{M}(D)\right]_{\varepsilon}$. We also have a corresponding distribution $\mathcal{P}_{(\mathcal{M},D,D',\varepsilon)}$ (Def. 12, abbreviated below as $\mathcal{P}$). Recall that the experiment of generating $\mathcal{P}$ is following: Toss a fair coin b. If $b=0$ the experiment outputs a sample o according to distribution $\left[\mathcal{M}(D)\right]_{\varepsilon}$, or otherwise outputs a sample o according to distribution $\mathcal{M}(D')$.

Let h^* and $R(h^*)$ be the Bayes classifier and the risk of the Bayes classifier for the distribution $\mathcal{P}$, respectively. Step 3 of algorithm $\mathcal{A}_{\mathcal{C}}^{\text{kNN}}$ (Figure 1) computes a kNN classifier $h_{k,n_1}^{\texttt{NN}}$ for distribution $\mathcal{P}$. Step 4 computes $\hat{R}_{n_2}(h_{k,n_1}^{\texttt{NN}})$, the testing risk of $h_{k,n_1}^{\texttt{NN}}$ with n_2 testing samples.

Because $\mathcal{M} \in \mathcal{C}$, the distribution of $\mathcal{M}(D')$ has density. Moreover, the distribution $\left[\mathcal{M}(D)\right]_{\varepsilon}$ *almost* has a density except at point $\perp$. By Chapter 11.2 of [15], the density assumption was needed to avoid problems caused by training points having equal distances to testing points (i.e., so that each point has exactly *k-nearest neighbors*). For the point $\perp$, we could define the distance from it to any other points as infinity, so at point $\perp$ the distance tie problem does not appear even without the density assumption. This means we could still use the result from Theorem 2. Thus, Theorem 2's condition suffices. By Theorem 2, when the sample size parameter n_1 is large enough, we have that

$$\Pr[|R(h_{k,n_1}^{\texttt{NN}}) - R(h^*)| > \alpha] \leqslant 2e^{-n_1\alpha^2/(72c_d^2)}.$$

Recall $n_1 = n/2$, defined in Step 1, Fig. 1. Set $2e^{-n_1\alpha^2/(72c_d^2)} = \beta/2$. Rearranging the inequality, with probability at least $1-\beta/2$,

$$|R(h_{k,n_1}^{\texttt{NN}}) - R(h^*)| \leqslant 12c_d\sqrt{\ln(4\beta)/n} \tag{5}$$

Plug the above inequality into Lemma 1, we have that for every $\delta_{D,D'}$ with respect to the $(\mathcal{M}, D, D', \varepsilon)$ and its estimate $\delta'_{D,D'}$ (defined in Step 5, Fig. 1)

$$|\delta'_{D,D'} - \delta_{D,D'}| \leqslant 24e^{\varepsilon}c_d\sqrt{\ln(4\beta)/n} + 2e^{\varepsilon}\sqrt{\ln(4/\beta)/n},$$

which completes the proof. □

Proof of Corollary 3. The proof is just to plug in Corollary 2 into the Theorem 5. So we have, with probability at least $1-\beta$,

$$\alpha \leqslant 24e^{\varepsilon}c_d\sqrt{\ln(8tm/\beta)/n} + 2e^{\varepsilon}\sqrt{\ln(8tm/\beta)/n}.$$

□

Definition 17 (The Laplacian vector query mechanism $\mathcal{M}_{L,\varepsilon}$)**.** *Let $\mathcal{M}_{L,\varepsilon}$ denote the DP vector query using Laplacian mechanism, which takes a d-dimensions real vector $x \in \mathsf{R}^d$ as input, samples a d-dimensions noise vector $v \sim Lap(\varepsilon)^d$ according to Laplace distribution[19], and then returns $x+v$ as the mechanism's output. $\mathcal{M}_{L,\varepsilon}$ is $(\varepsilon,0)$-differential private [28] if the L1 distance between any input x and any of its neighbor is 1.*

Definition 18 (The Gaussian vector query mechanism $\mathcal{M}_{G,\varepsilon,\delta}$)**.** *Let $\mathcal{M}_{G,\varepsilon,\delta}$ denote the DP vector query using Gaussian mechanism, which takes a d-dimensions real vector $x \in \mathsf{R}^d$ as input, samples a d-dimensions noise vector $v \sim \mathcal{N}(0, 2\varepsilon^{-2}\log(1.25/\delta))^d$ according to Gaussian distribution[20], and then returns $x+v$ as the mechanism's output. $\mathcal{M}_{G,\varepsilon,\delta}$ is (ε,δ)-differential private [28] if the L2 distance between any input x and any of its neighbor is 1.*

Lemma 2. *Let $\mathcal{M}_{L,\varepsilon}$ be the vector query mechanism defined in Definition 17 with dimension $d=1$. Let $\delta(\varepsilon')$ be the optimal δ (Def. 5) with respect to the tuple $(\mathcal{M}_{L,\varepsilon}, \varepsilon')$. $\delta(\varepsilon')$ satisfies the following equality*

$$\delta'(\varepsilon') = \begin{cases} 1 - e^{-\frac{1}{2}(\varepsilon-\varepsilon')} & \varepsilon' \in [0,\varepsilon] \\ 0 & \varepsilon' \geqslant \varepsilon. \end{cases} \tag{6}$$

Proof. Note that for a given ε, the optimal δ of $\mathcal{M}_{L,\varepsilon}$ is the maximum optimal δ among all neighboring pair given such ε. It only depends on the L1 distance between its input x and x's neighbor but not depends on what x is. The larger the L1 distance between x and its neighbor, the larger the optimal delta with respect to such neighboring pair. For $\mathcal{M}_{L,\varepsilon}$ with dimension $d=1$, it is obvious the the DP-spectrum with respect to $(D,D')=(0,1)$ equals to the DP-spectrum of $\mathcal{M}_{L,\varepsilon}$. So we only computed the optimal δ with respect to the tuple $(\mathcal{M}_{L,\varepsilon}, \varepsilon, 0, 1)$.

By Definition 5, we have

$$\delta(\varepsilon') = \max(\max_{\mathcal{S}\subseteq\mathcal{O}} \Pr[\mathcal{M}_{L,\varepsilon}(D) \in \mathcal{S}] - e^{\varepsilon'}\Pr[\mathcal{M}_{L,\varepsilon}(D') \in \mathcal{S}], 0),$$

where $\mathcal{O} = \texttt{Range}(\mathcal{M}_{L,\varepsilon})$.

For $\varepsilon' \geqslant \varepsilon$, by the differential privacy definition shown in Definition 3, we know

$$\max_{\mathcal{S}\subseteq\mathcal{O}} \Pr[\mathcal{M}_{L,\varepsilon}(D) \in \mathcal{S}] - e^{\varepsilon'}\Pr[\mathcal{M}_{L,\varepsilon}(D') \in \mathcal{S}] \leqslant 0,$$

19. The Laplace distribution (centered at 0) with parameter λ is the distribution with probability density function: $Lap(x \mid \lambda) = \frac{\lambda}{2}\exp\left(-\lambda|x|\right)$. We use $Lap(\lambda)$ to denote the Laplace distribution with parameter λ.

20. The Gaussian distribution with expectation 0 and variance σ^2 is the distribution with probability density function: $\mathcal{N}(x|\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{x^2}{2\sigma^2}\right)$.We use $\mathcal{N}(0,\sigma^2)$ to denote the Gaussian distribution with expectation 0 and variance σ^2

so that

$$\delta(\varepsilon') = 0.$$

Now we turn to the case $\varepsilon' < \varepsilon$. We first recall the probability density function of $\mathcal{M}_{L,\varepsilon}(D)$

$$\Pr[\mathcal{M}_{L,\varepsilon}(D) = x] = \frac{\varepsilon}{2} e^{-\varepsilon|x|},$$

where $x \in \mathsf{R}$. Similarly, the probability density function of $\mathcal{M}_{L,\varepsilon}(D')$ is

$$\Pr[\mathcal{M}_{L,\varepsilon}(D') = x] = \frac{\varepsilon}{2} e^{-\varepsilon|x-1|},$$

where $x \in \mathsf{R}$.

For $\varepsilon' < \varepsilon$,

$$\begin{aligned}
\delta(\varepsilon') &= \max(\max_{\mathcal{S}\subseteq\mathcal{O}} \Pr[\mathcal{M}_{L,\varepsilon}(D) \in \mathcal{S}] - e^{\varepsilon'} \Pr[\mathcal{M}_{L,\varepsilon}(D') \in \mathcal{S}], 0) \\
&= \max_{\mathcal{S}\subseteq\mathcal{O}} \Pr[\mathcal{M}_{L,\varepsilon}(D) \in \mathcal{S}] - e^{\varepsilon'} \Pr[\mathcal{M}_{L,\varepsilon}(D') \in \mathcal{S}] \\
&= \int_{-\infty}^{\infty} \max(0, \Pr[\mathcal{M}_{L,\varepsilon}(D) = x] - e^{\varepsilon'} \Pr[\mathcal{M}_{L,\varepsilon}(D') = x]) dx
\end{aligned} \tag{7}$$

Denote $x_+ \in \mathsf{R}$ such that $e^{-\varepsilon|x_+|} - e^{\varepsilon'} e^{-\varepsilon|x_+-1|} = 0$. The function $\Pr[\mathcal{M}_{L,\varepsilon}(D) = x] - e^{\varepsilon'} \Pr[\mathcal{M}_{L,\varepsilon}(D') = x]$ has only one zero, that is x_+. For all $x \leqslant x_+$, $\Pr[\mathcal{M}_{L,\varepsilon}(D) = x] - e^{\varepsilon'} \Pr[\mathcal{M}_{L,\varepsilon}(D') = x] \geqslant 0$, otherwise $\Pr[\mathcal{M}_{L,\varepsilon}(D) = x] - e^{\varepsilon'} \Pr[\mathcal{M}_{L,\varepsilon}(D') = x] < 0$. One can show

$$x_+ = \frac{1}{2}(1 - \frac{\varepsilon'}{\varepsilon}).$$

Plug in the equation 7, we have

$$\begin{aligned}
\delta(\varepsilon') &= \int_{-\infty}^{x_+} \Pr[\mathcal{M}_{L,\varepsilon}(D) = x] - e^{\varepsilon'} \Pr[\mathcal{M}_{L,\varepsilon}(D') = x] dx \\
&= \int_{-\infty}^{x_+} \frac{\varepsilon}{2}(e^{-\varepsilon|x|} - e^{\varepsilon'} e^{-\varepsilon|x-1|}) dx \\
&= 1 - e^{-\frac{1}{2}(\varepsilon-\varepsilon')},
\end{aligned}$$

where the last step is by integration. □

Lemma 3. *Let $\mathcal{M}_{G,\varepsilon,\delta}$ be the vector query mechanism defined in Definition 18 with dimension $d = 1$. Let $\delta(\varepsilon')$ be the optimal δ (defined in Def. 5) with respect to the tuple $(\mathcal{M}_{G,\varepsilon,\delta}, \varepsilon')$. $\delta(\varepsilon')$ satisfies the following equality*

$$\delta(\varepsilon') = \frac{1}{2}[1 + erf(\frac{x_+}{\sigma\sqrt{2}}) - e^{\varepsilon'}(1 + erf(\frac{x_+ - 1}{\sigma\sqrt{2}}))],$$

where $\sigma^2 = \frac{2\log(1.25/\delta)}{\varepsilon^2}$, $\varepsilon' > 0$, $x_+ = \frac{1}{2}(1 - 2\sigma^2\varepsilon')$ and $erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-s^2} ds$ (the standard error function.)

Proof. Due to the same reason as we explained in the proof of Lemma 2, since the DP-spectrum with respect to $(D, D') = (0, 1)$ equals to the DP-spectrum of $\mathcal{M}_{G,\varepsilon,\delta}$, we only computed the optimal δ with respect to the tuple $(\mathcal{M}_{L,\varepsilon}, \varepsilon, 0, 1)$.

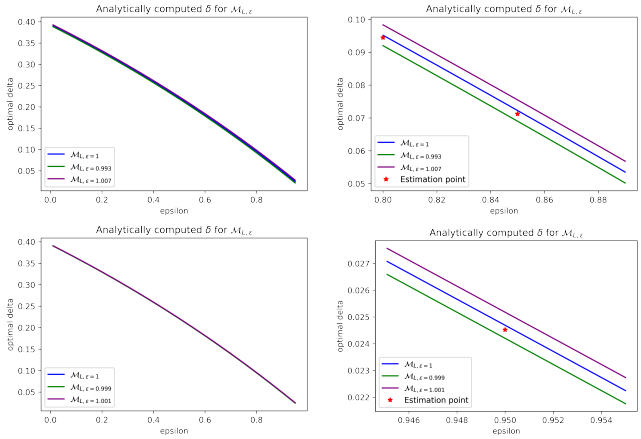


Figure 10: Application 2: verify implementation of $\mathcal{M}_{L,\varepsilon}$ mechanism, by checking which ε, δ trade-off curve the implementation falls under. Different curves represent $\mathcal{M}_{L,\varepsilon}$ with different amount of added noise.

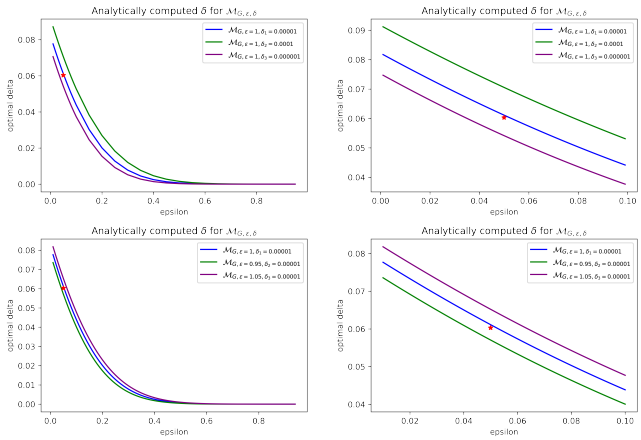


Figure 11: Application 2: verify the mechanism $\mathcal{M}_{G,\varepsilon,\delta}(\varepsilon = 1, \delta = 0.00001)$ is correctly implemented

By Definition 5, we have

$$\delta(\varepsilon') = \max(\max_{\mathcal{S}\subseteq\mathcal{O}} \Pr[\mathcal{M}_{G,\varepsilon,\delta}(D) \in \mathcal{S}] - e^{\varepsilon'} \Pr[\mathcal{M}_{G,\varepsilon,\delta}(D') \in \mathcal{S}], 0),$$

where $\mathcal{O} = \texttt{Range}(\mathcal{M}_{G,\varepsilon,\delta})$.

We then recall the probability density function of $\mathcal{M}_{G,\varepsilon,\delta}(D)$

$$\Pr[\mathcal{M}_{G,\varepsilon,\delta}(D) = x] = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}},$$

where $x \in \mathsf{R}$. Similarly, the probability density function of $\mathcal{M}_{G,\varepsilon,\delta}(D')$ is

$$\Pr[\mathcal{M}_{G,\varepsilon,\delta}(D') = x] = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-1)^2}{2\sigma^2}},$$

where $x \in \mathsf{R}$.

$x_+ = \frac{1}{2}(1 - 2\sigma^2\varepsilon')$ is the value such that $\Pr[\mathcal{M}_{G,\varepsilon,\delta}(D) = x_+] - e^{\varepsilon'} \Pr[\mathcal{M}_{G,\varepsilon,\delta}(D') = x_+] = 0$. The function $\Pr[\mathcal{M}_{G,\varepsilon,\delta}(D) = x] - e^{\varepsilon'} \Pr[\mathcal{M}_{G,\varepsilon,\delta}(D') = x]$ has only one zero, that is x_+. For all $x \leqslant x_+$, $\Pr[\mathcal{M}_{G,\varepsilon,\delta}(D) = x] - e^{\varepsilon'} \Pr[\mathcal{M}_{G,\varepsilon,\delta}(D') = x] \geqslant 0$, otherwise $\Pr[\mathcal{M}_{G,\varepsilon,\delta}(D) = x] - e^{\varepsilon'} \Pr[\mathcal{M}_{G,\varepsilon,\delta}(D') = x] < 0$.

Now we have, for all $\varepsilon' > 0$,

$$\begin{aligned}
\delta(\varepsilon') &= \max(\max_{\mathcal{S}\subseteq\mathcal{O}} \Pr[\mathcal{M}_{G,\varepsilon,\delta}(D) \in \mathcal{S}] - e^{\varepsilon'} \Pr[\mathcal{M}_{G,\varepsilon,\delta}(D') \in \mathcal{S}], 0) \\
&= \max_{\mathcal{S}\subseteq\mathcal{O}} \Pr[\mathcal{M}_{G,\varepsilon,\delta}(D) \in \mathcal{S}] - e^{\varepsilon'} \Pr[\mathcal{M}_{G,\varepsilon,\delta}(D') \in \mathcal{S}] \\
&= \int_{-\infty}^{\infty} \max(0, \Pr[\mathcal{M}_{G,\varepsilon,\delta}(D) = x] - e^{\varepsilon'} \Pr[\mathcal{M}_{G,\varepsilon,\delta}(D') = x]) dx \\
&= \int_{-\infty}^{x_+} \Pr[\mathcal{M}_{G,\varepsilon,\delta}(D) = x] - e^{\varepsilon'} \Pr[\mathcal{M}_{G,\varepsilon,\delta}(D') = x] dx \\
&= \int_{-\infty}^{x_+} \Pr[\mathcal{M}_{G,\varepsilon,\delta}(D) = x] - e^{\varepsilon'} \int_{-\infty}^{x_+} \Pr[\mathcal{M}_{G,\varepsilon,\delta}(D') = x] \\
&= (\frac{1}{2} + \frac{1}{2} erf(\frac{x_+}{\sigma\sqrt{2}})) - e^{\varepsilon'} (\frac{1}{2} + \frac{1}{2} erf(\frac{x_+ - 1}{\sigma\sqrt{2}})) \\
&= \frac{1}{2}[1 + erf(\frac{x_+}{\sigma\sqrt{2}}) - e^{\varepsilon'} (1 + erf(\frac{x_+ - 1}{\sigma\sqrt{2}})),
\end{aligned}$$

which completes the proof. □

3. More Experiments on Verifying Mechanism Implementation

Finally, for mechanisms for which the privacy spectrum can be analytically computed, e.g., Laplace (Lemma 2) and Gaussian (Lemma3) our verification can be even more accurate. To do so, we first generate several analytically computed (ε, δ) curves for $\mathcal{M}_{L,\varepsilon}$, w.r.t. added noise that guarantees at least $(\varepsilon, \delta = 0)$-DP, for $\varepsilon = 0.999, 1, 1.001$. We see (Fig. 10) that the ε, δ trade-off of the implementation is the closest to the analytically computed curve generated by mechanism $\mathcal{M}_{L,\varepsilon}$ with noise according to $\varepsilon = 1$, which is a good indication that in fact our implementation satisfies $\varepsilon = 1$. This same technique also applies to, e.g., the Gaussian mechanism (Fig. 11).