Boosting and Differential Privacy

Cynthia Dwork Guy N. Rothblum* Salil Vadhan

Abstract

Boosting is a general method for improving the accuracy of learning algorithms. We use boosting to construct *privacy-preserving synopses* of the input database. These are data structures that yield, for a given set $\mathcal{Q}$ of queries over an input database, reasonably accurate estimates of the responses to every query in $\mathcal{Q}$. Given a *base synopsis generator* that takes a distribution on $\mathcal{Q}$ and produces a "weak" synopsis that yields "good" answers for a majority of the weight in $\mathcal{Q}$, our *Boosting for Queries* algorithm obtains a synopsis that is good for all of $\mathcal{Q}$. We ensure privacy for the rows of the database, but the boosting is performed on the *queries*.

We provide the first base synopsis generator for sets of arbitrary low-sensitivity queries, *i.e.*, queries whose answers do not vary much under the addition or deletion of a single row.

Boosting is an iterative method. In our Boosting for Queries algorithm, each iteration incurs a certain privacy loss. In analyzing the cumulative privacy loss over many iterations, we obtain a bound on the *expected* privacy loss from a single ε-differentially private mechanism. Combining this with evolution of confidence arguments from the literature, we get a fresh perspective – and stronger bounds – on the expected cumulative privacy loss due to multiple mechanisms, each of which provides ε-differential privacy or one of its relaxations, and each of which operates on (potentially) different, adaptively chosen, databases.

We can also view a database as a training set in a learning algorithm, where each row corresponds to an element in the training set. Given the power and prevalence of boosting, it is natural to search for boosting techniques that preserve the privacy properties of the base learner. We present a differentially private boosting technique, in which privacy comes at little additional cost in accuracy. We call this *Boosting for People*, since database rows corresponding to individual people are the elements of interest.

*Center for Computational Intractability and Department of Computer Science, Princeton University, 35 Olden Street, Princeton, NJ 08544. rothblum@alum.mit.edu. Research supported by NSF Grant CCF-0832797 and by a Computing Innovation Fellowship. Part of this work was done while the author was visiting Microsoft Research.

1 Background and Summary of Results

Boosting. Boosting is a general and widely used method for improving the accuracy of learning algorithms. (See [30] for an excellent survey.) Given a training set of labeled examples, $\{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$, where each x_i is drawn from an underlying distribution D on a universe X, and $y_i \in \{+1, -1\}$, a learning algorithm produces a hypothesis $h : X \to \{+1, -1\}$. Ideally, h will "describe" not just the given samples, but also the underlying distribution. The goal of boosting is to convert a weak learner, which produces a hypothesis that does just a little better than random guessing, into a strong, or very accurate, learner. Many boosting algorithms share the following basic structure. First, an initial (typically uniform) probability distribution is imposed on the sample set. Computation then proceeds in rounds. In each round t: (1) the base learner is run on the current distribution D_t, producing a classification hypothesis h_t; and (2) the hypotheses $h_1, \ldots, h_t$ are used to re-weight the samples, defining D_{t+1}. The process halts either after a predetermined number of rounds or when an appropriate combining of the hypotheses is determined to be sufficiently accurate. The main design decisions are how to modify the probability distribution from one round to the next, and how to combine the hypotheses $\{h_t\}_{t=1,\ldots,T}$ to form a final output hypothesis.

Differential Privacy. Differential privacy is a notion of privacy tailored to private data analysis, where the goal is to learn information about the population as a whole, while protecting thea privacy of each individual. (See the surveys [10, 9].) Roughly speaking, differential privacy ensures that the system will behave in essentially the same fashion, independent of whether any individual opts in to, or out of, the database. Here, "behaves essentially the same way" means that the probability distribution over outputs of an analysis, where the probability space is the coin flips of the privacy mechanism, is essentially the same, independent of the presence or absence of any individual.

Early results on differential privacy showed how to accurately answer small to moderate numbers of *counting queries* of the form "How many rows in the database satisfy property P?" [15, 13]. Specifically, any set $\mathcal{Q}$ of counting queries could be answered in a differentially private manner with an accuracy of roughly $\sqrt{|\mathcal{Q}|}$, so for a database of size n, we can obtain nontrivial accuracy (namely, errors of magnitude $o(n)$) if $|\mathcal{Q}|$ is sufficiently smaller than n^2. (For simplicity, throughout the introduction, we hide dependence on parameters other than $|\mathcal{Q}|$ and n.) In [13, 11], similar bounds were obtained for arbitrary *low-sensitivity* queries, that is queries whose output does not change much when one item is added or removed from the database.

A remarkable result of Blum, Ligett, and Roth [5] shows that differential privacy is possible even in cases when the number of counting queries is much larger than n^2. Specifically, given a set $\mathcal{Q}$ of *counting queries*, they show how to answer all the queries in $\mathcal{Q}$ within an error of roughly $n^{2/3} \cdot \log^{1/3} |\mathcal{Q}|$, which provides nontrivial accuracy provided that $|\mathcal{Q}|$ is sufficiently smaller than 2^n. In fact, they also provide a compact representation of all of these answers in the form of a *synthetic database*. This is a data structure that "looks like" a database, in that its rows are drawn from the same universe $\mathcal{X}$ from which the database rows are drawn. The synthetic database produced by the mechanism in [5] has the property that, when appropriately scaled, the responses on the synthetic database to all the queries in $\mathcal{Q}$ approximate the answers to the same queries on the original database. Dwork *et al.* [14] improved the running time to $\text{poly}(|\mathcal{X}|, |\mathcal{Q}|)$, where $\mathcal{X}$ is the universe of the database rows, and achieved an incomparable accuracy bound of roughly $\sqrt{n} \cdot |\mathcal{Q}|^{o(1)}$.

Note that a compact representation for answers to a set $\mathcal{Q}$ of queries on a database x need

not be in the form of a synthetic database; it can be an arbitrary data structure, which can be queried for any $q \in \mathcal{Q}$ and return an approximation to $q(x)$. We refer to such a data structure as a *synopsis* of the database. General privacy-preserving synopses are of interest because they may be easier to construct than privacy-preserving synthetic databases. For example, there are stronger hardness results for constructing synthetic databases than are known for general privacy-preserving synopses [14, 34].

Summary of Results. Our principal result is a technique for generating privacy-preserving synopses for *any* set of low-sensitivity queries (not just counting queries). This is achieved by a novel use of boosting, together with the construction of an appropriate base synopsis generator.

Boosting for Queries. We introduce the notion of *boosting for queries*, where the items on which the boosting algorithm operates are the database queries, *i.e.,* the functions or analyses that the analyst wishes to evaluate on the database. Given a base synopsis generator that takes a distribution on $\mathcal{Q}$ and produces a "weak" synopsis that yields "good" answers for a majority of the weight in $\mathcal{Q}$, we wish to "boost" it to obtain a synopsis that is good for all of $\mathcal{Q}$. Although the boosting is performed over the queries, the privacy is still for the rows of the database. The privacy challenges in boosting for queries come from the fact that each row in the database affects the answers to all the queries. We present a privacy-preserving boosting for queries algorithm, which preserves the accuracy of the base synopsis generator up to a small additional error. The running time of the boosting procedure depends quasi-linearly on the number $|\mathcal{Q}|$ of queries and on the running time of the base synopsis generator. (In particular, it is independent of the data universe size $|\mathcal{X}|$.) This suggests a new avenue for constructing efficient and accurate privacy-preserving mechanisms, analogous to the approach enabled by boosting in the machine learning literature: an algorithm designer can tackle the (potentially much easier) task of constructing a weak privacy-preserving base synopsis generator, and automatically obtain a stronger mechanism.

Base Synopsis Generators for Arbitrary Low-Sensitivity Queries and for Counting Queries. We provide a base synopsis generator for sets of arbitrary low-sensitivity queries. Applying boosting to it, we get the first privacy-preserving synopsis construction for arbitrary low-sensitivity queries. The accuracy of our mechanism is roughly $\sqrt{n} \cdot \log^2 |\mathcal{Q}|$. The running time of our base synopsis generator (and hence its boosted version) is large, namely $\text{poly}(|\mathcal{Q}|, |\mathcal{X}|^n)$, where n is the size of the database.

For the special case of counting queries we can use a base synopsis generator from [14], and obtain the same accuracy as above (improving on the bound of $\sqrt{n} \cdot |\mathcal{Q}|^{o(1)|}$ from [14]) with a running time of $\text{poly}(|\mathcal{Q}|, |\mathcal{X}|)$. We note that the final mechanism of [14] is obtained by an iterative improvement to their base synopsis generator, but does not draw upon methods from the boosting literature (as we do).

Bounding Expected Privacy Loss and Composition Theorems. As noted above, boosting is an iterative technique. Each iteration of our algorithm provides ε-differential privacy, for some ε. We obtain a bound on the *expected*, as opposed to worst-case, privacy loss from a single ε-differentially private mechanism. Combining this with *evolution of confidence* arguments from the literature [8, 15], we get a fresh perspective – and new bounds – on the expected cumulative privacy loss due to multiple (say k) mechanisms, each providing ε-differential privacy or one of its relaxations (see Section 2), and each operating on (potentially) different, adaptively chosen, databases. Roughly speaking, privacy will deteriorate as $\sqrt{k}\varepsilon + k\varepsilon^2$, rather than the worst-case $k\varepsilon$ known in the literature [13].

Boosting for People. As previously done in [4, 20], we can also view the input database as a training set in a learning algorithm, where each row corresponds to an element in the training set. It is natural to try to combine learning and differential privacy: use learning theory to know what to compute on a database to understand the underlying population, and use the techniques for differential privacy to do this in a privacy-protective fashion, with small distortion when possible. We present a differentially private boosting technique, in which privacy comes at little additional cost in accuracy. We call this *Boosting for People*, since rows corresponding to the data of individual people are the elements of interest.

See Appendix A for a discussion of further related work.

2 Preliminaries and Definitions

We write $[n]$ for the set $\{1, 2, \ldots n\}$. Throughout the paper, we work with discrete probability spaces. Sometimes we will describe our algorithms as sampling from continuous distributions, but these should always be discretized to finite precision in some standard way (which we do not specify for sake of readability). For a discrete distribution (or random variable) X taking values in a set S, we denote by $x \leftarrow X$ the experiment of selecting $x \in S$ according to the distribution X. The *support* of X is denoted $\mathrm{Supp}(X) = \{x : \Pr[X = x] > 0\}$. A function $f : \mathcal{N} \to \mathbb{R}^+$ is *negligible* if for every constant c, $f(\kappa) < 1/\kappa^c$ for sufficiently large κ (i.e. $f(\kappa) = \kappa^{-\omega(1)}$). We write $\nu(\kappa)$ for an unspecified function that is negligible in κ.

In this work we deal with (non-interactive) methods for releasing information about a database. For a given database x, a (randomized) non-interactive database access mechanism $\mathcal{M}$ computes an output $\mathcal{M}(x)$ that can later be used to reconstruct information about x. We will be concerned with mechanisms $\mathcal{M}$ that are *private* according to various privacy notions described below.

We think of a database x as a multiset of *rows*, each from a data universe X. Intuitively, each row contains the data of a single individual. We will often view a database of size n as a tuple $x \in X^n$ for some $n \in \mathcal{N}$ (the number of individuals whose data is in the database). We treat n as public information throughout.

We say databases x, x' are *adjacent* if they differ only in one row, meaning that we can obtain one from the other by deleting one row and adding another. I.e. databases are adjacent if they are of the same size and their edit distance is 1. To handle worst case pairs of databases, our probabilities will be over the random choices made by the privacy mechanism.

Definition 2.1 (Differential Privacy [13])**.** A randomized algorithm $\mathcal{M}$ is *ε-differentially private* if for all pairs of adjacent databases x, x', and for all sets $S \subseteq \mathrm{Supp}(\mathcal{M}(x)) \cup \mathrm{Supp}(\mathcal{M}(x'))$

$$\Pr[\mathcal{M}(x) \in S] \leq e^{\varepsilon} \cdot Pr[\mathcal{M}(x') \in S],$$

where the probabilities are over the coin flips of the algorithm $\mathcal{M}$.

Intuitively, this captures the idea that no individual's data has a large effect on the output distribution of the mechanism. Typically, we think of ε as a small constant. A basic example of a differentially private algorithm is the *Laplace mechanism* [13], which yields differentially private approximations to real-valued functions. Specifically, for a real-valued function f, the *(global) sensitivity of f* is the maximal absolute difference in its values on adjacent databases: $\max_{\text{adjacent } x,x'} |f(x) - f(x')|$. The *Laplace distribution* $\mathrm{Lap}(t)$ has density function $h(y) \propto e^{-y/t}$, has mean 0 and standard deviation t. We usually refer to the Laplace distribution over integers.

Dwork *et al.* [13] showed that if a real-valued function f has global sensitivity at most s then the function $f(x) + \mathrm{Lap}(s/\varepsilon)$ is ε-differentially private. See [10, 9] for more properties and results concerning differential privacy.

We will also consider two relaxations of differential privacy, which allow us to ignore events of very low probability.

Definition 2.2 ((ε, δ)-Probabilistic Differential Privacy [22, ?])**.** A randomized algorithm $\mathcal{M}$ satisfies (ε, δ)-*probabilistic differential privacy* if for all databases x, we can divide $\mathrm{Range}(\mathcal{M})$ into two sets $\mathcal{S} = \mathcal{S}_x$ and $\bar{\mathcal{S}}$ such that $\Pr[\mathcal{M}(x) \in \bar{\mathcal{S}}] \leq \delta$, and for all x' adjacent to x, and all $S \subseteq \mathcal{S}$: $\Pr[\mathcal{M}(x) \in S] \leq e^{\varepsilon} Pr[\mathcal{M}(x') \in S]$ and $\Pr[\mathcal{M}(x') \in S] \leq e^{\varepsilon} Pr[\mathcal{M}(x) \in S]$, where the probabilities are over the coin flips of the algorithm $\mathcal{M}$.

Definition 2.3 ((ε, δ)-Differential Privacy [11])**.** A randomized algorithm $\mathcal{M}$ gives (ε, δ)-*differential privacy* if for all pairs of adjacent databases x and x' and all $S \subseteq \mathrm{Range}(\mathcal{M})$, $\Pr[\mathcal{M}(x) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(x') \in S] + \delta$, where the probabilities are over the coin flips of the algorithm $\mathcal{M}$.

Observe that ε-differential privacy implies (ε, δ)-probabilistic differential privacy, which implies (ε, δ)-differential privacy. There are simple examples showing the converse implications do not hold.

Auxiliary parameters and synopsis generators. Often our privacy mechanism $\mathcal{M}$ will take some auxiliary parameters w as input, in addition to the database x. For example, w may specify a query q_w on the database x, or a collection $\mathcal{Q}_w$ of queries. The Mechanism $\mathcal{M}(w, x)$ might (respectively) respond with a differentially private approximation to $q_w(x)$ or to some or all of the queries in $\mathcal{Q}_w$. We say that a mechanism $\mathcal{M}(\cdot, \cdot)$ satisfies ε-differential privacy if for every w, $\mathcal{M}(w, \cdot)$ satisfies ε-differential privacy; and analogously for the other notions of privacy.

Another example of a parameter that may be included in w is a *security parameter* κ to govern how small $\delta = \delta(\kappa)$ should be. That is, $\mathcal{M}(\kappa, \cdot)$ should be $(\varepsilon, \delta(\kappa))$ differentially private for all κ. Typically, and throughout this paper, we require that δ be a negligible function in κ, i.e. $\delta = \kappa^{-\omega(1)}$. Thus, we think of δ as being cryptographically small, whereas ε is typcially thought of as a moderately small constant.

In the case where the auxiliary parameter w specifies a collection $\mathcal{Q}_w = \{q : X^n \to \mathbb{R}\}$ of queries, we call the mechanism $\mathcal{M}$ a *synopsis generator*. A synopsis generator outputs a (differentially private) synopsis A which can be used to compute answers to all the queries in $\mathcal{Q}_w$. I.e., we require that there exists a reconstruction procedure R such that for each input v specifying a query $q_v \in \mathcal{Q}_w$, the reconstruction procedure outputs $R(A, v) \in \mathbb{R}$. Typically, we will that with high probability $\mathcal{M}$ produces a synopsis A s.t. the reconstruction procedure, using A, computes accurate answers. I.e., for all or most (weighted by some distribution) of the queries $q_v \in \mathcal{Q}_w$, the error $|R(A, v) - q_v(x)|$ will be bounded. We will occasionally abuse notation and refer to the reconstruction procedure taking as input the actual query q (rather than some representation v of it), and outputting $R(A, q)$.

Divergence. For the next section, it will be useful to rephrase the privacy notions above in terms of distance measures between distributions. In the fractional quantities below, if the denominator is 0, then we define the value of the fraction to be infinite (the numerators will always be positive). For two discrete random variables Y and Z, their *KL divergence* (i.e. *relative entropy*) is defined to be

$$D(Y||Z) \stackrel{\mathrm{def}}{=} \mathrm{E}_{y \leftarrow Y}\left[\ln \frac{\Pr[Y = y]}{\Pr[Z = y]}\right].$$

It is known that $D(Y||Z) \geq 0$, with equality iff Y and Z are identically distributed. (However, D is not symmetric, does not satisfy the triangle inequality, and can even be infinite, specifically when $\mathrm{Supp}(Y)$ is not contained in $\mathrm{Supp}(Z)$.) We can obtain a worst-case analogue of KL divergence by taking a maximum instead of an expectation (analogous to how min-entropy relates to Shannon entropy):

$$D_\infty(Y||Z) \stackrel{\text{def}}{=} \max_{y \in \mathrm{Supp}(Y)} \left[\ln \frac{\Pr[Y=y]}{\Pr[Z=y]} \right] = \max_{S \subseteq \mathrm{Supp}(Y)} \left[\ln \frac{\Pr[Y \in S]}{\Pr[Z \in S]} \right].$$

We refer to $D_\infty(Y||Z)$ as the *max-divergence* of Y and Z. This is a rather brittle measure, in that a change in even a small portion of probability space can affect $D_\infty(Y||Z)$ dramatically. Thus it is natural to allow ourselves to discard a small fraction of the probability space, leading to *δ-approximate max-divergence*, which we define by :

$$D^\delta_\infty(Y||Z) \stackrel{\text{def}}{=} \max_{S \subseteq \mathrm{Supp}(Y): \Pr[Y \in S] > \delta} \left[\ln \frac{\Pr[Y \in S] - \delta}{\Pr[Z \in S]} \right].$$

Observe that differential privacy and (ε, δ)-differential privacy can be be formulated in terms of these measures: (i) A randomized algorithm $\mathcal{M}$ gives ε-differential privacy iff for all pairs of adjacent databases x and x', we have $D_\infty(\mathcal{M}(x)||\mathcal{M}(x')) \leq \varepsilon$. And (ii) A randomized algorithm $\mathcal{M}$ gives (ε, δ)-differential privacy iff for all pairs of adjacent databases x and x', we have $D^\delta_\infty(\mathcal{M}(x)||\mathcal{M}(x')) \leq \varepsilon$.

One other distance measure that will be useful is *statistical distance* between two random variables Y and Z, defined as

$$\Delta(Y,Z) \stackrel{\text{def}}{=} \max_S |\Pr[Y \in S] - \Pr[Z \in S]|.$$

We say that Y and Z are *δ-close* if $\Delta(Y,Z) \leq \delta$.

The following reformulations of approximate max-divergence in terms of exact max-divergence and statistical distance will be convenient for us:

Lemma 2.1. *1. $D^\delta_\infty(Y||Z) \leq \varepsilon$ if and only if there exists a random variable Y' such that $\Delta(Y,Y') \leq \delta$ and $D_\infty(Y'||Z) \leq \varepsilon$.*

2. We have both $D^\delta_\infty(Y||Z) \leq \varepsilon$ and $D^\delta_\infty(Z||Y) \leq \varepsilon$ if and only if there exist random variables Y', Z' such that $\Delta(Y,Y') \leq \delta/(e^\varepsilon+1)$, $\Delta(Z,Z') \leq \delta/(e^\varepsilon+1)$, and $D_\infty(Y'||Z') \leq \varepsilon$.

The above relations can be seen as information-theoretic analogues of the "dense model theorems" of [18, 33, 28, 27].

Proof. 1. Suppose there exists Y' δ-close to Y such that $D_\infty(Y||Z) \leq \varepsilon$. Then for every S,

$$\Pr[Y \in S] \leq \Pr[Y' \in S] + \delta \leq e^\varepsilon \cdot \Pr[Z \in S] + \delta,$$

and thus $D^\delta_\infty(Y||Z) \leq \varepsilon$.

Conversely, suppose that $D^\delta_\infty(Y||Z) \leq \varepsilon$. Let $S = \{y : \Pr[Y=y] > e^\varepsilon \cdot \Pr[Z=y]\}$. Then

$$\sum_{y \in S} (\Pr[Y=y] - e^\varepsilon \cdot \Pr[Z=y]) = \Pr[Y \in S] - e^\varepsilon \cdot \Pr[Z \in S] \leq \delta.$$

Moreover, if we let $T = \{y : \Pr[Y = y] < \Pr[Z = y]\}$, then we have

$$\begin{aligned}\sum_{y \in T}(\Pr[Z = y] - \Pr[Y = y]) &= \sum_{y \notin T}(\Pr[Y = y] - \Pr[Z = y]) \\ &\geq \sum_{y \in S}(\Pr[Y = y] - \Pr[Z = y]) \\ &\geq \sum_{y \in S}(\Pr[Y = y] - e^{\varepsilon} \cdot \Pr[Z = y])/\end{aligned}$$

Thus, we can obtain Y' from Y by lowering the probabilities on S and raising the probabilities on T to satisfy:

(a) For all $y \in S$, $\Pr[Y' = y] = e^{\varepsilon} \cdot \Pr[Z = y] < \Pr[Y = y]$.
(b) For all $y \in T$, $\Pr[Y = y] \leq \Pr[Y' = y] \leq \Pr[Z = y]$.
(c) For all $y \notin S \cup T$, $\Pr[Y' = y] = \Pr[Y = y] \leq e^{\varepsilon} \cdot \Pr[Z = y]$.

Then $D_{\infty}(Y'||Z) \leq \varepsilon$ by inspection, and

$$\Delta(Y, Y') = \Pr[Y \in S] - \Pr[Y' \in S] = \Pr[Y \in S] - e^{\varepsilon} \cdot \Pr[Z \in S] \leq \delta.$$

2. Suppose there exist random variables Y' and Z' as stated. Then, for every set S,

$$\begin{aligned}\Pr[Y \in S] &\leq \Pr[Y' \in S] + \frac{\delta}{e^{\varepsilon} + 1} \\ &\leq e^{\varepsilon} \cdot \Pr[Z' \in S] + \frac{\delta}{e^{\varepsilon} + 1} \\ &\leq e^{\varepsilon} \cdot \left(\Pr[Z \in S] + \frac{\delta}{e^{\varepsilon} + 1}\right) + \frac{\delta}{e^{\varepsilon} + 1} \\ &= e^{\varepsilon} \cdot \Pr[Z \in S] + \delta.\end{aligned}$$

Thus $D_{\infty}^{\delta}(Y||Z) \leq \varepsilon$, and by symmetry, $D_{\infty}^{\delta}(Z||Y) \leq \varepsilon$.

Conversely, given Y and Z such that $D_{\infty}^{\delta}(Y||Z) \leq \varepsilon$ and $D_{\infty}^{\delta}(Z||Y) \leq \varepsilon$, we proceed similarly to Part 1. However, instead of simply decreasing the probability mass of Y on S to obtain Y' and eliminate the gap with $e^{\varepsilon} \cdot Z$, we also increase the probability mass of Z on S. Specifically, for every $y \in S$, we'll take

$$\Pr[Y' = y] = e^{\varepsilon} \cdot \Pr[Z' = y] = \frac{e^{\varepsilon}}{1 + e^{\varepsilon}} \cdot (\Pr[Y = y] + \Pr[Z = y]) \in [e^{\varepsilon} \cdot \Pr[Z = y], \Pr[Y = y]].$$

This also implies that for $y \in S$, we have:

$$\Pr[Y = y] - \Pr[Y' = y] = \Pr[Z' = y] - \Pr[Z = y] = \frac{\Pr[Y = y] - e^{\varepsilon} \cdot \Pr[Z = y]}{e^{\varepsilon} + 1},$$

and thus

$$\alpha \stackrel{\text{def}}{=} \sum_{y \in S}\big(\Pr[Y = y] - \Pr[Y' = y]\big) = \sum_{y \in S}\big(\Pr[Z' = y] - \Pr[Z = y]\big) = \frac{\Pr[Y \in S] - e^{\varepsilon} \cdot \Pr[Z \in S]}{e^{\varepsilon} + 1} \leq \frac{\delta}{e^{\varepsilon} + 1}.$$

Similarly on the set $S' = \{y : \Pr[Z = y] > e^{\varepsilon} \cdot \Pr[Y = y]\}$, we can decrease the probability mass of Z and increase the probability mass of Y by a total of some $\alpha' \leq \delta/(e^{\varepsilon} + 1)$ so that for every $y \in S'$, we have $\Pr[Z' = y] = e^{\varepsilon} \cdot \Pr[Y' = y]$.

If $\alpha = \alpha'$, then we can take $\Pr[Z' = y] = \Pr[Z = y]$ and $\Pr[Y' = y] = \Pr[Y = y]$ for all $y \notin S \cup S'$, giving $D_{\infty}(Y||Z) \leq \varepsilon$ and $\Delta(Y, Y') = \Delta(Z, Z') = \alpha$. If $\alpha \neq \alpha'$, say $\alpha > \alpha'$, then we need to still increase the probability mass of Y' and decrease the mass of Z' by a total of $\beta = \alpha - \alpha'$ on points outside of $S \cup S'$ in order to ensure that the probabilities sum to 1. That is, if we try to take the "mass functions" $\Pr[Y' = y]$ and $\Pr[Z' = y]$ as defined above, then while we do have the property that for every y, $\Pr[Y' = y] \leq e^{\varepsilon} \cdot \Pr[Z' = y]$ and $\Pr[Z' = y] \leq e^{\varepsilon} \cdot \Pr[Y' = y]$ we also have $\sum_y \Pr[Y' = y] = 1 - \beta$ and $\sum_y \Pr[Z' = y] = 1 + \beta$. However, this means that if we let $R = \{y : \Pr[Y' = y] < \Pr[Z' = y]\}$, then

$$\sum_{y \in R} \big(\Pr[Z' = y] - \Pr[Y' = y]\big) \geq \sum_{y} \big(\Pr[Z' = y] - \Pr[Y' = y]\big) = 2\beta.$$

So we can increase the probability mass of Y' on points in R by a total of β and decrease the probability mass of Z' on points in R by a total of β, while retaining the property that for all $y \in R$, $\Pr[Y' = y] \leq \Pr[Z' = y]$. The resulting Y' and Z' have the properties we want: $D_{\infty}(Y', Z') \leq \varepsilon$ and $\Delta(Y, Y'), \Delta(Z, Z') \leq \alpha$.

□

3 Composition Theorems

In this section, we provide general results about the composition of differentially private mechanisms. There are several reasons for studying composition (analogous to the reasons that researchers have studied the composition of cryptographic protocols):

1. Composition can be used for the modular design of complex private mechanisms from simpler ones.

2. Composition models repeated use of the *same* mechanism on the *same database*; we want to be assured that its privacy guarantees will not degrade too much.

3. Composition models the interaction between many *different* privacy mechanisms. If Alice's data is used in many differentially private data releases over her lifetime, involving different databases and different mechanisms, we still would like to assure her that her privacy will not be compromised too much.

Previous composition results for differential privacy have primarily been concerned with the first two items. Here we introduce a form of composition that captures the very general setting suggested in Item 3, and prove composition results for it. Moreover, we revisit the "evolution of confidence" arguments due to Dinur, Dwork, and Nissim [8, 15] and show that, for achieving (ε, δ) differential privacy of k-fold composition, the privacy parameter ε degrades as $k\varepsilon^2 + \sqrt{k}\varepsilon$ rather than linearly as $k\varepsilon$. Previously this was shown only for specific mechanisms, and only when all k applications were on the same database.

3.1 Modeling Composition

We want to model composition where the adversary can adaptively affect the databases being input to future mechanisms, as well as the queries to those mechanisms. We do this by introducing a differentially private analogue of the "left or right" notion of security for encryption schemes, due to Bellare, Desai, Jokipii, and Rogaway [3]. Let $\mathbb{M}$ be a family of database access mechanisms. (For example $\mathbb{M}$ could be the set of all ε-differentially private mechanisms.) For a probabilistic adversary A, we consider two experiments, Experiment 0 and Experiment 1, defined as follows.

k-fold Composition Experiment b for mechanism family $\mathbb{M}$ and adversary A:[1] For $i = 1, \ldots, k$:

1. A outputs two adjacent databases x_i^0 and x_i^1, a mechanism $\mathcal{M}_i \in \mathbb{M}$, and parameters w_i.
2. A receives $y_i \leftarrow \mathcal{M}_i(w_i, x_{i,b})$.

We allow the adversary A above to be stateful throughout the experiment, and thus it may choose the databases, mechanisms, and the parameters adaptively depending on the outputs of previous mechanisms. We define A's *view* of the experiment to be A's coin tosses and all of the mechanism outputs $(y_1, \ldots, y_k)$. (The x_i^j's, $\mathcal{M}_i$'s, and w_i's can all be reconstructed from these.)

For intuition, consider an adversary who always chooses x_i^0 to hold Bob's data and x_i^1 to differ only in that Bob's data is replaced with junk. Then experiment 0 can be thought of as the "real world," where Bob allows his data to be used in many data releases, and Experiment 1 as an "ideal world," where the outcomes of these data releases do not depend on Bob's data. Our definitions of privacy still require these two experiments to be "close" to each other, in the same way as required by the definitions of differential privacy. The intuitive guarantee to Bob is that the adversary "can't tell", given the output of all k mechanisms, whether Bob's data was ever used.

Definition 3.1. We say that the family $\mathbb{M}$ of database access mechanisms satisfies *ε-differential privacy under k-fold adaptive composition* if for every adversary A, we have $D_\infty(V^0||V^1) \leq \varepsilon$ where V^b denotes the view of A in k-fold Composition Experiment b above.

(ε, δ)-differential privacy under k-fold adaptive composition instead requires that $D_\infty^\delta(V^0||V^1) \leq \varepsilon$.

3.2 Composition Theorems

Speaking colloquially, it is already known that when we compose differentially private mechanisms "the epsilons and deltas add up" (cf. [13] for ε-differential privacy and [12, 26, 11] for (ε, δ)-differential privacy). This also extends to our general model of composition:

Theorem 3.1. *For every $\varepsilon, \delta \geq 0$ and $k \in \mathbb{N}$,*

1. *The family of ε-differentially private mechanisms satisfies $k\varepsilon$-differential privacy under k-fold adaptive composition.*

[1] We remark that allowing both a mechanism family $\mathbb{M}$ and auxiliary parameters w is redundant. The parameters w can be removed by expanding the family $\mathbb{M}$ to $\mathbb{M}' = \{\mathcal{M}(\cdot, w)\}_{\mathcal{M} \in \mathbb{M}, w}$, and conversely, we can use the parameters to collapse $\mathbb{M}$ to a single mechanism $\mathcal{M}^*(x, (\mathcal{M}, w))$ that outputs $\mathcal{M}(x, w)$ if $\mathcal{M} \in \mathbb{M}$ and outputs $\perp$ otherwise.

2. *The family of (ε, δ)-differentially private mechanisms satisfies $(k\varepsilon, k\delta)$-differential privacy under k-fold adaptive composition.*

Thus, if Bob's data is to be involved in k data releases over his lifetime, the above theorem suggests that he should require both ε and δ to be smaller than $1/k$. For δ, this is not problematic as we anyhow take δ to be very small (negligible). But requiring ε to be this small can cause a significant price in utility (e.g. expected error $\Theta(k)$ is incurred in the Laplace mechanism).

For specific mechanisms applied on a single database, there are "evolution of confidence" arguments due to Dinur, Dwork, and Nissim [8, 15] showing that the privacy parameter need only deteriorate like $\sqrt{k}$ if we are willing to tolerate a (negligible) loss in δ (for $k < 1/\varepsilon^2$). Here we generalize those arguments to arbitrary differentially private mechanisms, as well as to the general form of composition described above.

Our key technical contribution shows that a bound of ε on the *worst-case* privacy loss (as captured by max-divergence) implies a bound of $O(\varepsilon^2)$ on the *expected* privacy loss (as captured by KL divergence). The proof is in Appendix B.

Lemma 3.2. *Suppose that random variables Y and Z satisfy $D_\infty(Y||Z) \leq \varepsilon$ and $D_\infty(Z||Y) \leq \varepsilon$. Then $D(Y||Z) \leq \varepsilon \cdot (e^\varepsilon - 1)$.*

Note that $e^\varepsilon \leq 1 + 2\varepsilon$ for $\varepsilon \in [0, 1]$, so we get a bound of $D(Y||Z) \leq 2\varepsilon^2$. Next we apply concentration bounds to show that with high probability the "privacy loss" is close to the expectation (which, by the above, is $O(\varepsilon^2 k)$ rather than εk), and thus giving us (ε', δ) for privacy for $\varepsilon' \ll k\varepsilon$ for composing k ε-differentially private mechanisms. Due to the adaptivity of the adversary, the outputs of mechanisms $\mathcal{M}_1, \ldots, \mathcal{M}_k$ are not independent. Following [8, 15], we use Azuma's Inequality (see Appendix B) to establish the concentration we want.

For composition of (ε, δ)-differential privacy, we use the characterization of approximate max-divergence from Lemma 2.1 (Part 1) to reduce the analysis to the same situation as in the case of ε-differential privacy. This yields the following theorem (which includes the case of ε-differential privacy by setting $\delta = 0$):

Theorem 3.3. *For every $\varepsilon > 0, \delta, \delta' > 0$, and $k \in \mathbb{N}$, the class of (ε, δ)-differentially private mechanisms is $(\varepsilon', k\delta + \delta')$-differentially private under k-fold adaptive composition, for*

$$\varepsilon' = \sqrt{2k \ln(1/\delta')} \cdot \varepsilon + k \cdot \varepsilon\varepsilon_0,$$

where $\varepsilon_0 = e^\varepsilon - 1$.

4 Boosting for Queries

In this section we present a query-boosting algorithm for arbitrary low-sensitivity queries. The algorithm takes a weak, sometimes-accurate, differentially private *"base synopsis generator"* (i.e. one that outputs privacy-preserving answers to most of the mass of a given query distribution), and "boosts" it to get an always (or often) accurate synopsis generator. This is done while maintaining differential privacy. We begin with a more formal treatment of the setup, follow with an overview and then present the query-boosting algorithm in Figure 1 and its accuracy and privacy guarantees in Theorem 4.1.

The Setup. Fix a database size n, a data universe X and a query set $\mathcal{Q} = \{q : X^n \to \mathbb{R}\}$ of real-valued queries. Recall the framework outlined in the introduction. We are given a database $x \in X^n$ and a class $\mathcal{Q} = \{q : X^* \to \mathbb{R}\}$ of (arbitrary low sensitivity) queries. The sensitivity of the query family $\mathcal{Q}$ is denoted $\rho = \rho(n) = \max\{q \in Q, \text{ adjacent } x, x' \in X^n\}$ (the maximum over all queries in the family of their sensitivities). We wish to answer the queries in a differentially private manner. To do so, we are given a *base synopsis generator*. For any distribution D on the query set $\mathcal{Q}$, the output of base synopsis generator can be used for computing accurate answers for a *large fraction* of the queries (weighted by D).

Definition 4.1 ($(k, \lambda, \eta, \beta)$-base synopsis generator)**.** For a fixed database size n, data universe X and query set $\mathcal{Q}$, consider an synopsis generator $\mathcal{M}$, that takes as input k queries from the query family $\mathcal{Q}$ and outputs a synopsis. We say that $\mathcal{M}$ is a $(k, \lambda, \eta, \beta)$-*base synopsis generator* if for any distribution D on $\mathcal{Q}$, when $\mathcal{M}$ is activated on a database $x \in X^n$ and on k queries sampled independently from D, with all but β probability (over the k queries drawn from D and the coins of $\mathcal{M}$) the synopsis that $\mathcal{M}$ outputs is λ-accurate (w.r.t x) for a $(1/2 + \eta)$-fraction of mass of $\mathcal{Q}$ as weighted by D.

We will be interested in differentially private base synopsis generators. The goal is to *"boost"* such base synopsis generator into a strong synopsis generator that answers all or almost all of the queries (w.r.t. the same database x) accurately with all but very small probability, while still preserving (ε, δ)-differential privacy. Given these parameters, we would like to optimize the accuracy of the resulting synopsis generator (aiming to stay as close as possible to λ-accuracy). We let μ denote the additional error incurred by the resulting synopsis generator.

Overview. We use the base synopsis generator, running it repeatedly for T rounds, generating a synopsis $\mathcal{A}_t$ in round t. We maintain a distribution $\mathcal{D}$ on the queries (initially this distribution is uniform), and re-weight so that in each round queries for which the base synopsis generator failed to produce an accurate answer get higher probability. The objects $\mathcal{A}$ are combined by taking the median: given $\mathcal{A}_1, \ldots, \mathcal{A}_T$, the quantity $q(DB)$ is estimated by taking the approximate values for $q(DB)$ computed using each of the $\mathcal{A}_i$, and then computing their median. The algorithm will run for a fixed number T of rounds (roughly $T \in O(\log(|Q|))$.

There are "standard" methods for updating $\mathcal{D}$, for example, increasing the weight of poorly handled elements, in our case, queries, by a factor of e and decreasing the weight of well-handled elements by the same factor. However, we need to protect the privacy of the database rows, and a single database row can have a substantial effect on $\mathcal{D}$ if it makes the difference between being "well approximated" or "poorly approximated" for many queries. We therefore need to mitigate the effect of any database row. This is done by attenuating the re-weighting procedure. Instead of always using a fixed ratio either for increasing the weight (when the answer is "accurate") or decreasing it (when it is not), we set separate thresholds for "accuracy" and "inaccuracy". Queries for which the error is below or above these thresholds have their weight decreased or increased (respectively) by a fixed factor. For queries whose error lies between these two thresholds, we scale the (logarithm of the) weight change linearly. The attenuated scaling reduces the effect of any individual on a the re-weighting of any query. This is because any individual can only affect the true answer to a query, and thus also the accuracy of the base synopsis generator's output, by a small amount.

The larger the gap between the "accurate" and "inaccurate" thresholds, the smaller the effect of each individual on a query's weight can be. This means that larger gaps are better for privacy. For

accuracy, however, large gaps are bad. If the inaccuracy threshold is large, we can only guarantee that queries for which the base synopsis generator is very inaccurate have their weight increased during re-weighting. This degrades the accuracy guarantee of the boosted synopsis generator: it is roughly equal to the "inaccuracy" threshold.

The query-boosting algorithm is general. It can be used for any class of queries (not only counting queries) and any differentially private base synopsis generator. The running time is, to a large extent, inherited from the base synopsis generator, which need only be run roughly $\log |\mathcal{Q}|$ times (assuming it has constant advantage η over 1/2). The booster invests additional time that is quasi-linear in $|\mathcal{Q}|$ in the boosting process, and in particular its running time does not depend directly on the size of the data universe from which data items come.

Notation. Throughout the algorithm's operation, we keep track of several variables (explicitly or implicitly). Variables indexed by $q \in \mathcal{Q}$ hold information pertaining to query q in the query set. We run T rounds of boosting. Variables indexed by $t \in [T]$, usually computed in round t, will be used to construct the distribution D_{t+1} used for sampling in time period $t+1$. For a predicate P we use $[[P]]$ to denote 1 if the predicate is true and 0 if it is false.

Boosting for Queries$(k, \lambda, \eta, \rho, \mu, T)$
Given: database $x \in X^n$, query set $\mathcal{Q}$, where each $q \in \mathcal{Q}$ is a function $q : X^n \to \mathbb{R}$ with sensitivity at most ρ.
Initialize D_1 to be the uniform distribution over $\mathcal{Q}$.
For $t = 1, \ldots, T$:

1. Sample a sequence $S_t \subseteq \mathcal{Q}$ of k samples chosen independently and at random from D_t.
   Run the base synopsis generator to compute an answer data structure $\mathcal{A}_t : \mathcal{Q} \to \mathbb{R}$ that is w.h.p. accurate for at least $(1+\eta)/2$ of the mass of D_t.
2. Reweight the queries. For each $q \in \mathcal{Q}$:
   (a) If $\mathcal{A}_t$ is λ-accurate, then $a_{t,q} \leftarrow 1$
   If $\mathcal{A}_t$ is $(\lambda+\mu)$-inaccurate, then $a_{t,q} \leftarrow -1$
   Otherwise, let $d_{q,t} = |q(x) - \mathcal{A}_t(q)|$ be the error of $\mathcal{A}_t$ (between λ and $\lambda+\mu$) on q:
   $$a_{t,q} \leftarrow 1 - 2(d_{t,q} - \lambda)/\mu$$
   (b) $u_{t,q} \leftarrow \exp(-\alpha \cdot \sum_{j=1}^{t} a_{j,q})$, where $\alpha = 1/2 \ln((1+2\eta)/(1-2\eta))$.
3. Renormalize:
   $$Z_t \leftarrow \sum_{q \in \mathcal{Q}} u_{t,q}$$
   $$D_{t+1}[q] = u_{t,q}/Z_t$$

Output the final answer data structure $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_T)$. For $q \in \mathcal{Q}$:
$$\mathcal{A}(q) = \mathit{median}\{\mathcal{A}_1(q), \ldots, \mathcal{A}_T(q)\}$$

Figure 1: **Boosting for Queries** (a variant of AdaBoost [31])

Theorem 4.1 (Boosting for Queries)**.** *Let $\mathcal{Q}$ be a query family with sensitivity at most ρ. For an appropriate setting of parameters, and with $T = O(\log |Q|/\eta^2)$ rounds, the algorithm of Figure 1 is an accurate and differentially private query-boosting algorithm:*

1. *When instantiated with a $(k, \lambda, \eta, \exp(-\kappa))$-base synopsis generator, the output of the boosting algorithm gives $(\lambda + \mu)$-accurate answers to* all *the queries in $\mathcal{Q}$ with probability at least $1 - T \cdot \exp(-\kappa)$, where $\mu = O((\log^{3/2} |Q| \cdot \sqrt{k} \cdot \sqrt{\kappa} \cdot \rho)/(\varepsilon \cdot \eta^4))$.*
2. *If the base synopsis generator is $(\varepsilon_{base}, \delta_{base})$-differentially private, then the boosting algorithm is $((\varepsilon + T \cdot \varepsilon_{base}), T \cdot (\exp(-\kappa) + \delta_{base}))$-differentially private.*

Proof of Theorem 4.1. We prove accuracy and privacy.

Accuracy. We will show that after T rounds of boosting, with all but $T \cdot \exp(-\kappa)$ probability, the answers to all but a $\exp(-\eta^2 \cdot T)$-fraction of the queries are $(\lambda+\mu)$-accurate. The proof follows the structure of Adaboost's accuracy analysis in [31].

Lemma 4.2. *After T rounds of boosting, with all but $T \cdot \exp(-\kappa)$ probability, the answers to all but an $\exp(-\eta^2 \cdot T)$-fraction of the queries are $(\lambda + \mu)$-accurate.*

Proof. In the last round of boosting, we have:

$$D_{T+1}[q] \quad = \quad \frac{u_{T,q}}{Z_T} \tag{1}$$

Take $a^+_{t,q}$ to be -1 if $\mathcal{A}_j$ is $(\lambda+\mu)$-inaccurate for q, and 1 otherwise. For all rounds t, observe that $a^+_{t,q} \geq a_{t,q}$: if $a^+_{t,q} = 1$ then $\mathcal{A}_j$ is $(\lambda + \mu)$-accurate, and so $a_{t,q} \leq 1$. This implies that:

$$u^+_{T,q} \triangleq e^{-\alpha \sum_{t=1}^{T} a^+_{t,q}} \leq e^{-\alpha \sum_{t=1}^{T} a_{t,q}} = u_{T,q} \tag{2}$$

Combining Equations (1) and (2), for all $q \in \mathcal{Q}$:

$$D_{T+1}[q] \geq \frac{u^+_{T,q}}{Z_T} \tag{3}$$

Now we turn to examining the predicate $[[\mathcal{A}$ is $(\lambda + \mu)$-inaccurate for $q]]$, which tests whether the booster's output $\mathcal{A}$ is accurate for a query q. If this predicate is 1, then it must be the case that the majority of $\{\mathcal{A}_j\}_{j=1}^{T}$ are $(\lambda+\mu)$-inaccurate, as otherwise their median would be $(\lambda+\mu)$-accurate. We conclude that:

$$\begin{aligned} [[\mathcal{A} \text{ is } (\lambda+\mu)\text{-inaccurate for } q]] \quad &\Rightarrow \quad \sum_{t=1}^{T} a^+_{t,q} \leq 0 \\ &\Leftrightarrow \quad e^{-\alpha \sum_{t=1}^{T} a^+_{t,q}} \geq 1 \\ &\Leftrightarrow \quad u^+_{T,q} \geq 1 \end{aligned}$$

Since $u^+_{T,q} \geq 0$, we conclude that:

$$[[\mathcal{A} \text{ is } (\lambda+\mu)\text{-inaccurate for } q]] \leq u^+_{T,q}$$

Using the above together with Equation (3):

$$\begin{aligned}\frac{1}{|\mathcal{Q}|}\cdot\sum_{q\in\mathcal{Q}}[[\mathcal{A}\text{ is }(\lambda+\mu)\text{-inaccurate for }q]] &\leq \frac{1}{|\mathcal{Q}|}\cdot\sum_{q\in\mathcal{Q}}u^{+}_{T,q}\\ &\leq \frac{1}{|\mathcal{Q}|}\cdot\sum_{q\in\mathcal{Q}}D_{T+1}[q]\cdot Z_T\\ &\leq \frac{Z_T}{|\mathcal{Q}|}\end{aligned}$$

Thus the following claim completes the proof:

Claim 4.3. *In round t of boosting, with all but $t\cdot\exp(-\kappa)$ probability:*

$$Z_t\leq\exp(-\eta^2\cdot t)\cdot|\mathcal{Q}|$$

Proof. Take $a^{-}_{t,q}$ to be 1 if $\mathcal{A}_t$ is λ-accurate on q, and -1 otherwise. It is always the case that $a_{t,q}\geq a^{-}_{t,q}$ (it always holds that $a_{t,q}\geq -1$, and if $a^{-}_{t,q}=1$, then $\mathcal{A}_j$ is λ-accurate, and so $a_{t,q}=1$).

Throughout, we use η to denote the accuracy guarantee of the base synopsis generator. The guarantee that the base synopsis generator has is that in every round t, with all but $\exp(-\kappa)$ probability, it is λ-accurate for at least a $(1/2+\eta)$-fraction of the mass of the distribution D_t. We assume throughout that this accuracy holds in every round (ignoring the $\exp(-\kappa)$ failure probability until the end). In round t, we will examine the following quantity:

$$r_t\triangleq\sum_{q\in\mathcal{Q}}D_t[q]\cdot a^{-}_{t,q}$$

which measures the "success" of the base synopsis generator in round t. Here we still use the "stricter" notion of "success" as λ-accuracy. If a $(1/2+\eta_t)$-fraction of the mass of D_t is computed with λ-accuracy, then $r_t\geq 2\eta_t$. We have a guaranteed lower bound on η (a parameter of the base synopsis generator), which also yields a lower bound $r_t\geq 2\eta$ in every round t. Now observe also that for $t\in[T]$:

$$\begin{aligned}Z_t &= \sum_{q\in\mathcal{Q}}u_{t,q}\\ &= \sum_{q\in\mathcal{Q}}u_{t-1,q}\cdot e^{-\alpha\cdot a_{t,q}}\\ &= \sum_{q\in\mathcal{Q}}Z_{t-1}\cdot D_t[q]\cdot e^{-\alpha\cdot a_{t,q}}\\ &\leq \sum_{q\in\mathcal{Q}}Z_{t-1}\cdot D_t[q]\cdot e^{-\alpha\cdot a^{-}_{t,q}}\\ &\leq Z_{t-1}\cdot\sum_{q\in\mathcal{Q}}D_t[q]\cdot\left(\left(\frac{1+a^{-}_{t,q}}{2}\right)\cdot e^{-\alpha}+\left(\frac{1-a^{-}_{t,q}}{2}\right)\cdot e^{\alpha}\right)\\ &= Z_{t-1}\cdot(1/2)\cdot\left[(e^{\alpha}+e^{-\alpha})+r_t\cdot(e^{-\alpha}-e^{\alpha})\right]\\ &\leq Z_{t-1}\cdot(1/2)\cdot\left[(e^{\alpha}+e^{-\alpha})+2\eta\cdot(e^{-\alpha}-e^{\alpha})\right]\end{aligned}$$

Following [31], we optimize by choosing

$$\alpha = (1/2)\ln\left(\frac{1+2\eta}{1-2\eta}\right)$$

and get that

$$Z_t \leq (\sqrt{1-4\eta^2})^t \cdot |\mathcal{Q}|$$

which implies

$$Z_t \leq \exp(-2\eta^2 \cdot t) \cdot |\mathcal{Q}|$$

□

□

Privacy. The proof of privacy considers adjacent databases x and x'. In each of the T rounds of boosting, fixing the past answers $\mathcal{A}_1, \ldots, \mathcal{A}_t$, we use D_{t+1} to denote the next round's distribution computed using database x and D'_{t+1} to denote the next round's distribution computed using database x'. For the various quantities computed by the algorithm in round t using database x, such as $u_{t,q}$ or Z_t, we use ${u_{t,q}}'$ or Z'_t to denote their counterparts computed using database x'.

In Claim 4.4 below, we show that (fixing the previously computed $\mathcal{A}_j$'s), for adjacent x, x', the max-divergence of the distributions D_{t+} and D'_{t+1} is bounded.

Claim 4.4. *Let x and x' be adjacent databases. After $t \in [T]$ rounds of boosting, fix $\mathcal{A}_1, \ldots, \mathcal{A}_t$. Let D_{t+1} and D'_{t+1} be the corresponding distributions for the $(t+1)$-th round of boosting. Then $D_\infty(D_{t+1}||D'_{t+1}) \leq 4\alpha \cdot T \cdot \rho/\mu$.*

Proof. For every $q \in \mathcal{Q}$, if we take $d_{q,t} = |q(x) - \mathcal{A}_t(q)|$ and $d'_{q,t} = |q(x') - \mathcal{A}_t(q)|$, then we get that $|d_{q,t} - d'_{q,t}| \leq \rho$. This implies that $|a_{t,q} - a'_{t,q}| \leq 2\rho/\mu$, and so:

$$e^{-2\alpha \cdot T \cdot \rho/\mu} \leq u_{t,q}/u'_{t,q} \leq e^{2\alpha \cdot T \cdot \rho/\mu}$$

and thus (since $Z_t = \sum_{q \in \mathcal{Q}} u_{t,q}$ and $Z'_t = \sum_{q \in \mathcal{Q}} u'_{t,q}$):

$$e^{-2\alpha \cdot T \cdot \rho/\mu} \leq Z_t/Z'_t \leq e^{2\alpha \cdot T \cdot \rho/\mu}$$

The claim follows from the above because $D_{t+1}[q] = u_{t,q}/Z_t$ and $D'_{t+1}[q] = u'_{t,q}/Z'_t$. □

We model the databases accesses throughout the T rounds of boosting as an instance of the composition model of Section 3.1. Each such database access takes as input the synopses generated in previous rounds: there are T executions of the base synopsis generator, a mechanism which is $(\varepsilon_{base}, \delta_{base})$-differentially private, and we also sample $k \cdot T$ times from the distributions $\{D_t\}_{t=1}^T$, each such sample is a $(4\alpha \cdot T \cdot \rho/\mu)$-differentially private mechanism (by Claim 4.4). Using the composition theorems (Theorems 3.1 and 3.3), we conclude that the boosting algorithm in its entirety is: $(\varepsilon_{boost}, \delta_{boost})$-differentially private, where

$$\varepsilon_{boost} = O((T \cdot \varepsilon_{base}) + \sqrt{T \cdot k \cdot \kappa} \cdot ((\alpha \cdot T \cdot \rho)/\mu))$$

$$\delta_{boost} = T \cdot (\delta_{base} + \exp(-\kappa)))$$

To get the parameters claimed in the theorem statement we can take:

$$\mu = O((T^{3/2} \cdot \sqrt{k} \cdot \sqrt{\kappa} \cdot \alpha \cdot \rho)/\varepsilon)$$

The algorithm sets $\alpha = (1/2)\ln((1+2\eta)/(1-2\eta)) = O(\eta)$. For accuracy, we need the number of rounds to be $T = O(\log|Q|/\eta^2)$. This yields the noise bound claimed in the theorem. □

5 Applications of Boosting for Queries

In this section we detail applications of the query boosting algorithm. We construct base synopsis generators using a generalization argument due to [14], see Section 5.1. In Section 5.2 we use this bound to construct base synopsis generators for arbitrary low-sensitivity queries (with high running time) and counting queries (with better running time and accuracy). Plugging these base synopsis generators into the boosting for queries algorithm yields new boosted mechanisms for answering large numbers of low-sensitivity queries, we detail the parameters that are obtained in Section 5.3.

5.1 A Generalization Bound

We have a distribution D over a large set $\mathcal{Q}$ of queries to be approximated. If we wanted accurate and differentially private answers to *all* the queries in $\mathcal{Q}$, then the standard approach of adding (say Gaussian) noise would yield error roughly proportional to $\sqrt{|\mathcal{Q}|}$. If, however, we only want accurate answers to *most* of the queries in $\mathcal{Q}$ (weighted by the distribution D), we can (in some settings) employ a generalization argument of [14]. They show that if a small enough synopsis (a synthetic database or some other data structure) gives good enough approximations to the answers of a *randomly selected* subset $S \subset \mathcal{Q}$ of queries sampled by D, then with high probability (over the choice of S) it also gives good approximations to the answers to *most* queries in $\mathcal{Q}$ (weighted by D). As they showed, this observation can lead to significantly better approximations.

We use $R(y,q)$ to denote the answer given by the synopsis y (when used as input for the reconstruction procedure) on query q. Formally, we say that a synopsis y *λ-fits* a database x w.r.t a set S of queries if $\max_{q\in S}|R(y,q) - q(x)| \leq \lambda$. The generalization bound shows that if y λ-fits x w.r.t a large enough (larger than $|y|$) randomly chosen set S of queries sampled from a distribution D, then w.h.p y λ-fits x for *most* of D's query mass.

The proof of Lemma 5.1 is a generalization to arbitrary distributions of the statement in [14] (that statement was made for the uniform distribution).

Lemma 5.1. [14] *Let $\mathcal{D}$ be an arbitrary distribution on a query set $\mathcal{Q} = \{q : X^* \to \mathbb{R}\}$. For all $m \in \mathcal{N}$, $\beta \in (0,1)$, $\eta \in [0, 1/2)$, take $a = 2(\log(1/\beta) + m)/(m \cdot (1-2\eta))$. Then with probability at least $1-\beta$ over the choice of $S \sim D^{a\cdot m}$, every synopsis y of size at most m bits that λ-fits x w.r.t. the query set S, also λ-fits x w.r.t. at least a $(1/2+\eta)$-fraction of D.*

Proof. Fix a randomly chosen set of queries $S \subset \mathcal{Q}$ chosen according to $\mathcal{D}^{a\cdot m}$. Examine an arbitrary m-bit synopsis y. Note that y is described by an m-bit string. Let us say y is *bad* if $|R(y,q) - q(x)| > \lambda$ for at least a $(\log(1/\beta)+m)/(a\cdot m)$ fraction of $\mathcal{D}$, meaning that $\Pr_{q\sim\mathcal{D}}[|R(y,q) - q(x)| > \lambda] \geq (\log(1/\beta)+m)/(a\cdot m)$.

In other words, y is bad if there exists a set $Q_y \subset \mathcal{Q}$ of fractional weight at least $(\log(1/\beta) + m)/(a\cdot m)$ such that $|R(y,q) - q(x)| > \lambda$ for $q \in Q_y$. For such a y, what is the probability that y

gives λ-accurate answers for *every* $q \in S$? This is exactly the probability that none of the queries in S is in Q_y, or

$$(1 - (\log(1/\beta) + m)/(a \cdot m))^{a \cdot m} \leq e^{-(\log(1/\beta)+m)} \leq \beta \cdot 2^{-m}$$

Taking a union bound over all 2^m possible choices for y, the probability that there exists an m-bit synopsis y that is accurate on all the queries in S but inaccurate on a set of fractional weight $(log(1/\beta) + m)/(a \cdot m)$ is at most β.

For $0 < \beta < 1$, to ensure that, with probability $1 - \beta$, there exists a synthetic database y that answers well on a $(1/2 + \eta)$ fraction of $\mathcal{Q}$, it is sufficient to have $(\log(1/\beta) + m)/(a \cdot m) < 1/2 - \eta$, or:

$$a > \frac{2(\log(1/\beta) + m)}{m \cdot (1 - 2\eta)}$$

□

5.2 Base Synopsis Generators

We now use the generalization bound to construct privacy-preserving base synopsis generators. Given a query distribution D, the idea is to sample a small set of queries from D, answer them by adding independent noise, and then output a synopsis (a synthetic database of bounded size) that "fits" the noisy answers. By Lemma 5.1, if the number of queries we sampled was large enough (larger than the synopsis size), this synopsis will also (w.h.p.) "accurately answer" a random query sampled from D. As mentioned above, here our synopses will be synthetic databases. We show how to find a database that fits the noisy answers for arbitrary queries (in large time), and for counting queries (more efficiently, following [14]). These base synopses appear in the theorems below.

Theorem 5.2 (Base Synopsis Generator for Arbitrary Queries)**.** *For any data universe X, database size n, and class $\mathcal{Q} : \{X^n \to \mathbb{R}\}$ of queries of sensitivity at most ρ, for any $\varepsilon, \kappa > 0$, there exists a $(k, \lambda, \eta = 1/3, \beta = \exp(-\kappa))$-base synopsis generator for Q, where*

$$k = O(n \cdot \log(|X|) \cdot \kappa), \lambda = \tilde{O}\left(\frac{\sqrt{n \cdot \log|X|} \cdot \rho \cdot \kappa^{3/2}}{\varepsilon}\right)$$

Moreover, this bases synopsis generator is $(\varepsilon, \exp(-\kappa))$-differentially private. Its running time is $|X|^n \cdot \text{poly}(n, \kappa, \log(1/\varepsilon))$.

Proof. The base synopsis generator will output a synopsis y, which is a (synthetic) database of size n, i.e. it is of bit length $m = n \cdot \log|X|$. The base synopsis generator gets a set S of $k = a \cdot m$ queries from $\mathcal{Q}$, where $a = O(\kappa)$ (the queries are sampled from some underlying distribution D). It answers each query $q \in S$, and adds independent Laplace noise of magnitude $O(\sqrt{k \cdot \kappa} \cdot \rho/\varepsilon)$ to each answer. The output is a function of these noisy answers (see below), and so, by Theorem 3.3, the base synopsis generator is $(\varepsilon, \exp(-\kappa))$-differentially private.

Let $\widehat{q(x)}$ be the collection of noisy answers computed above. The base synopsis generator enumerates over all $|X|^n$ databases of size n, and outputs the lexicographically first database y such that for every $q \in S$ we have $|q(y) - \widehat{q(x)}| \leq \lambda/2$. If no such database is found, it outputs $\perp$ instead (i.e. fails).

For accuracy, observe that with all but $\exp(-\kappa)/2$ probability all the k noise values added are of magnitude at most $O((\sqrt{k \cdot \kappa} \cdot \rho \cdot \kappa \cdot \log k)/\varepsilon) = \lambda/2$. Let us condition on this event (the "moderate noise" event). If the enumeration succeeds, then any y for which $|q(y) - q(x)| \leq \lambda/2$ for each of the query in S also λ-fits the input database x w.r.t. x. Using the generalization bound of Lemma 5.1, we conclude that with all but $\exp(-\kappa)/2$ probability (over the choice of S from D^k) the y that λ-fits x w.r.t S, also λ-fits x w.r.t. at least a $(1/2 + 1/3)$-fraction of D.

We still need to argue that such a y exists (otherwise the enumeration fails). This is because (conditioning on "moderate noise") the input database x is such a database. Thus this is indeed a $(k, \lambda, 1/3, \exp(-\kappa))$-base synopsis generator. The running time is dominated by enumerating the X^n possible databases to find y. □

Theorem 5.3 (Base Synopsis Generator for Counting Queries [14])**.** *For any data universe X, database size n, and class $\mathcal{Q} : \{X^n \to \mathbb{R}\}$ of counting queries (with sensitivity at most $1/n$), for any $\varepsilon, \kappa > 0$, there exists a $(k, \lambda, \eta = 1/3, \beta = \exp(-\kappa))$-base synopsis generator for Q, where*

$$k = \tilde{O}(n \cdot \log(|X|) \cdot \kappa / \log |Q|), \lambda = \tilde{O}\left(\frac{\sqrt{\log |Q|} + \sqrt{\log |X|} \cdot \kappa^{3/2}}{\varepsilon \cdot \sqrt{n}}\right)$$

Moreover, this bases synopsis generator is $(\varepsilon, \exp(-\kappa))$-differentially private. Its running time is $\text{poly}(|X|, n, \kappa, \log(1/\varepsilon))$.

Proof. The base synopsis generator will output a synopsis y, which is a (synthetic) database of size $n' = \tilde{O}(n/\log |Q|)$, i.e. it is of bit length $m = \tilde{O}(n \cdot \log |X| / \log |Q|)$. The base synopsis generator gets a set S of $k = a \cdot m$ queries from $\mathcal{Q}$, where $a = \tilde{O}(\kappa)$ (the queries are sampled from some underlying distribution D). The only difference from the base synopsis generator for general queries of Theorem 5.2 is that the size of the database that we output is smaller. The rest of the algorithm is very similar. It answers each query $q \in S$, and adds independent Laplace noise of magnitude $O(\sqrt{k \cdot \kappa} \cdot \rho/\varepsilon)$ and then find a database that fits these noisy answers (see below). By Theorem 3.3, the base synopsis generator is $(\varepsilon, \exp(-\kappa))$-differentially private.

We use the noisy answers to compute a database y of size n' such that for every $q \in S$ we have $|q(y) - \widehat{q(x)}| \leq 2\lambda/3$ (or fail if we find no such database). As before, with all but $\exp(-\kappa)/2$ probability all the k noise values added are of magnitude at most $O((\sqrt{k \cdot \kappa} \cdot \rho \cdot \kappa \cdot \log k)/\varepsilon) \leq \lambda/3$, and we condition on this event. Thus, if we a y that approximates the noisy answers as above, then y also λ-fits the input database x w.r.t. x. Using the generalization bound of Lemma 5.1, we conclude that with all but $\exp(-\kappa)/2$ probability (over the choice of S from D^k) a y that λ-fits x w.r.t S, also λ-fits x w.r.t. at least a $(1/2 + 1/3)$-fraction of D.

We need to argue that such a y of size n' exists (otherwise the enumeration fails). Here the existence of the database x itself is not sufficient, because it is of size $n > n'$. However, following [5] we can argue (using a sampling argument) that there *exists* a database x' of size $\tilde{O}(n/\log |Q|) \leq n'$ that $\sqrt{\log |Q|/n} < \lambda/3$-fits the database x. In particular, this database x' will $2\lambda/3$-fit the noisy answers, and this suffices for proving that the y we want exists.

The running time is dominated by the time required for finding y. This is done in [14] using linear programming in time $\text{poly}(|X|, n)$. □

5.3 Putting It Together

Plugging the base synopsis generators of Section 5.2 together into the query boosting algorithm, we obtain the following boosted algorithms for answering large sets of low-sensitivity queries:

Theorem 5.4 (Boosted Synopsis Generator for Arbitrary Queries)**.** *For any data universe X, database size n, and class $\mathcal{Q} : \{X^n \to \mathbb{R}\}$ of queries of sensitivity at most ρ, for any $\varepsilon, \kappa > 0$, there exists an $(\varepsilon, \exp(-\kappa))$-differentially private synopsis generator for Q. With all but $\exp(-\kappa)$ probability its answers are λ-accurate for every query in $\mathcal{Q}$, where*

$$\lambda = \tilde{O}\left(\frac{\sqrt{n \cdot \log|X|} \cdot \rho \cdot \log^{3/2}|\mathcal{Q}| \cdot \kappa^{3/2}}{\varepsilon}\right)$$

Its running time is $|X|^n \cdot |Q| \cdot \mathrm{poly}(n, \kappa, \log(1/\varepsilon))$.

Theorem 5.5 (Boosted Synopsis Generator for Counting Queries)**.** *For any data universe X, database size n, and class $\mathcal{Q} : \{X^n \to \mathbb{R}\}$ of counting queries (with sensitivity at most $1/n$), for any $\varepsilon, \kappa > 0$, there exists an $(\varepsilon, \exp(-\kappa))$-differentially private synopsis generator for Q. With all but $\exp(-\kappa)$ probability its answers are λ-accurate for every query in $\mathcal{Q}$, where*

$$\lambda = \tilde{O}\left(\frac{\sqrt{\cdot \log|X|} \cdot \log|\mathcal{Q}| \cdot \kappa^{3/2}}{\varepsilon \cdot \sqrt{n}}\right)$$

Its running time is $\mathrm{poly}(|X|, |Q|, n, \kappa, \log(1/\varepsilon))$.

6 Boosting for People

In this section we present a boosting algorithm that takes a "weak" somewhat-accurate differentially private base learner, and "boosts" it to get a much more accurate (and still differentially private) hypothesis.

The Setup. We assume we have access to a *privacy-preserving base learner*. The base learner gets k data items from a universe X, each labeled by ± 1, and outputs a classifier (or hypothesis $h : X \to \{\pm 1\}$). We associate each item with an individual's private data. The privacy guarantee of the base learner is $(\varepsilon_{base}, \delta_{base})$-differential privacy for changes of single items in the input set. The accuracy guarantee is below:

Definition 6.1 ((k, η, β)-Base Learner)**.** Let X be a universe, $C = \{c : X \to \pm 1\}$ a concept class, and L a learning algorithm that receives k labeled items, each in $X \times \{\pm 1\}$ and outputs a hypothesis $h : X \to \{\pm 1\}$.

We say that L is a (k, η, β)-base learner for C if for every $c \in C$, when the k items are drawn from a distribution D on X and labeled by c, with all but β probability over the choice of the items and L's coins, the hypothesis h labels at least a $(1/2 + \eta)$-fraction of the mass of D correctly (i.e. $\Pr_{(x,y)\sim D}[h(x) = c(x)] \geq 1/2 + \eta$).

The goal in "people-boosting" is, given such a base learner and a set of m labeled data items, to (i) guarantee (ε, δ)-differential privacy for ε and δ that are not much larger than $\varepsilon_{base}, \delta_{base}$ (respectively), and (ii) output a classifier that classifies correctly all but a γ-fraction of its input items (with all but very small probability). The goal (as it is in much of the boosting literature) is that this classifier should then generalize well to items that were not in the training set (by an

Occam's razor argument, assuming the base learner's classifiers are themselves "simple" enough to generalize).

Given these parameters, we would like to optimize the size m of the input dataset as a function of the desired parameters $\varepsilon, \delta, \gamma$ and the fixed parameters of the base learner $\varepsilon_{base}, \delta_{base}, k, \eta, \beta$. Other parameters of interest include the number of calls to the base learner (the number of rounds of boosting), the computational and space complexities, and more.

Overview and Challenges. A popular approach in the boosting literature is to run in iterative rounds, in each round "re-weighting" data items so that items that get classified incorrectly by the base learner get higher weight, and items that get classified correctly get their weight reduced (forcing the base learner to "pay attention" to improperly labeled data items). Often the weights of these re-weighted items are then normalized (implicitly or explicitly) to form a distribution.

In each round we sample a few (k) items according to the (re-weighted) distribution in that round, and give them as input to the base learner (we assume the base learner ensures good generalization error for samples of size k). The main challenge for privacy is to ensure (in a differentially private manner) that, in the re-weighted distributions generated by the boosting algorithm, no item's weight becomes too high. Note that the base learner's privacy guarantee is only with respect to small changes in its input. If one item is likely to occur many times in this input, then the base learner's privacy does not suffice to ensure that the presence or absence of this high-weight item will not significantly affect the output. If, however, we can maintain a "smoothness" property for our distribution, then no individual item should be likely to occur too many times in the base learner's input, and the privacy of the base learner will imply that the weak hypothesis it generates are differentially private.

There is a growing literature on such smooth boosting algorithms [32, 21, 17, 1], where no single example is assigned too much weight. However, note that smoothness of the distributions alone does not guarantee differential privacy. The "smoothness" of the distributions needs to be maintained in a differentially private matter. As an example, one difficulty is handling normalization: when we normalize, each item's weight can affect the weights of all other items, and arguing that the set of re-weighted weights is differentially private can become challenging. This is why we cannot use a known smooth boosting algorithms as a black box. We do, however, note that it remains entirely possible that adapting an existing or future smooth boosting algorithm to preserve privacy might yield a simpler or better solution.

We tackle these issues, adapting a variant of Freund and Schapire's Adaboost algorithm [31] to our setting. We present a smooth and privacy-preserving boosting algorithm. Smoothness is maintained by capping the weight of each item in the re-weighted distributions. We use a noisy weight cap to guarantee differential privacy. Normalization issues are handled by proving that, as long as we work only with smooth distributions, each data item only has a small effect on the normalization. This, in turn, means that we can upper-bound the *statistical distance* between re-weighted distributions on adjacent datasets. We show that drawing samples from statistically close distributions and feeding them to a differentially private mechanism ensures differential privacy. The boosting theorem is below. The explicit algorithm and analysis follow.

Theorem 6.1. *Let L be a base learner for a concept class C. Then for any $\gamma, \varepsilon^*, \kappa > 0$, when L is plugged into the boosting algorithm of Figure 2, for appropriate settings of the parameters (and appropriate constants hidden in the $O(\cdot)$ notation) the boosting algorithm guarantees:*

1. *If L is $(\varepsilon_{base}, \delta_{base})$-differentially private and we run T rounds of boosting, then the output of the boosting algorithm is $(\varepsilon^* + \varepsilon_{base}, T \cdot \delta_{base} \cdot (1 + e^{\varepsilon_{base}}))$-differentially private.*

2. *If L is a (k, η, β)-base learner, then the boosting algorithm runs for $T = O(1/\eta^2 \cdot \log(1/\gamma))$ rounds of boosting. For datasets of size at least*

$$m = O((T \cdot k + T \cdot \kappa/\varepsilon^*) \cdot (1/\eta^2) \cdot \log(1/\gamma)/\gamma)$$

it guarantees that with all but $(T \cdot \beta + \exp(-\kappa))$ probability, the output hypothesis has error at most γ.

The smoothness bound of the intermediate distributions is $1/\ell = O(1/(T \cdot k + T \cdot \kappa/\varepsilon^*))$

Notation. Throughout the algorithm's operation, we keep track of several variables (explicitly or implicitly). Variables indexed by $i \in [m]$ will be used to hold information pertaining to item i in the input set. We will run T rounds of boosting. Variables indexed by $t \in [T]$, usually computed in round t, will be used to construct the distribution D_{t+1} used for sampling in time period $t + 1$. For a predicate P we use $[[P]]$ to denote 1 if the predicate is true and 0 if it is false.

Proof of Theorem 6.1. We want to prove both privacy and utility. Utility will come from the Adaboost analysis, together with the fact that not too many of the participants' weights are ever capped. Privacy will come from the noise added, the base learner's privacy, and the smoothness of the maintained distributions.

Correctness and Utility Analysis. To argue utility, we need to analyze how the weight capping we introduce can be incorporated into the analysis of standard (uncapped) boosting (for which we follow [31]). We assume throughout that in every round (i) the base learner "succeeds" (i.e. output a hypothesis that is correct for a $(1/2 + \eta)$ fraction of that round's distribution), (ii) it holds that $\ell \leq \widehat{\ell}_t \leq 3\ell$, and (iii) the noise in testing the termination condition of Step 3 is at most $\gamma \cdot m/4$. This is the case with all but $O(T \cdot \beta + \exp(-\Omega(\varepsilon \cdot \ell)) + \exp(-\Omega(\varepsilon \cdot \gamma \cdot m))$ probability. We also require that $m > 3\ell$.

First, to argue that in every round t the "smoothness threshold" w_t exists, is unique, and can be computed efficiently, observe that for small positive w_t the sum $\sum_{i=1}^{m} \min(v_{i,t-1} \cdot \exp(-\alpha \cdot c_{i,t})/w_t, 1)$ equals $m > \widehat{\ell}$, for increasing w_t going to infinity, the sum converges to 0, and the sum strictly decreases as w_t grows.

By items whose weight has been capped in round t, we mean items i for which $1 < v_{i,t-1} \cdot \exp(-\alpha \cdot c_{i,t})/w_t$ (and so their weight is decreased by the min operator). To bound the error, we consider in every round t the set U_t of items $i \in [m]$ whose weight has not been *"capped"* in the rounds $1, \ldots, t$ (where $U_0 = \Phi$). For each item, we consider the "uncapped" ratios of its weight change:

$$u_{i,t} = e^{-\alpha \cdot c_{i,t}}$$

And the "capped" ratio $r_{i,t} = v_{i,t}/v_{i,t-1} \leq u_{i,t}$. We also define a normalizer for the re-weighting in round t:

$$Z_t = \sum_i D_t[i] \cdot r_{i,t}$$

In particular we get that:

$$D_{t+1}[i] = \frac{D_t[i] \cdot r_{i,t}}{Z_t}$$

Boosting for People$(\gamma, m, \ell, \varepsilon^*, \varepsilon_{base}, \eta, \gamma, \kappa)$
Given: $DB = (x_1, y_1), \ldots, (x_m, y_m)$, where $x_i \in X$, $y_i \in \{-1, 1\}$
Initialize: $\varepsilon = \varepsilon^*/2T$ and $\forall i \in [m] : D_1[i] = v_{i,0} = 1/m$
For $t = 1, \ldots, T$:

1. Sample (with replacement) a set C_t of k labeled items according to the distribution D_t.
2. Use the set C_t and the base learner to train a classifier $h_t : X \to \{\pm 1\}$.
3. Check whether the current accumulated hypothesis is "good enough". Compute:

$$err_t \leftarrow |\{i \in [m] : \text{majority}(h_1(x_i), \ldots, h_t(x_i)) = y_i\}|$$

   if $(err_t + \text{Lap}(1/\varepsilon))/m \leq \gamma/2$, then terminate the loop. Otherwise, continue.
4. For $i \leftarrow 1 \ldots m$: $c_{i,t} \leftarrow h_t(x_i) \cdot y_i \in \{\pm 1\}$.
5. Privacy-preserving "smooth" re-weighting: choose $\widehat{\ell_t} \leftarrow 2\ell + \text{Lap}(1/\varepsilon)$, terminate if $\widehat{\ell_t} < \ell$ or if $\widehat{\ell_t} \geq m$.

   Otherwise, compute the unique $w_t > 0$ s.t.:

$$\widehat{\ell_t} = \sum_{i=1}^{m} \min(v_{i,t-1} \cdot \exp(-\alpha \cdot c_{i,t})/w_t, 1)$$

   and set the new capped weight:

$$v_{i,t} \leftarrow \min(v_{i,t-1} \cdot \exp(-\alpha \cdot c_{i,t})/w_t, 1)$$

   and update:

$$D_{t+1}[i] \leftarrow v_{i,t} \Big/ \left(\sum_i v_{i,t} \right) = v_{i,t} \Big/ \widehat{\ell_t}$$

After terminating in round $t \in [T]$, the final classifier is:

$$H(x) = \text{majority}\{h_1(x_i), \ldots, h_t(x_i)\}$$

Figure 2: **Boosting for People**

The proof of utility proceeds by bounding the number of errors made by the classifier after round T as a function of the number of uncapped items and of $\prod_{t=1}^{T} Z_t$. This is done in Claim 6.2. In Claims 6.3 we argue that the Z_t's are all noticeable smaller than 1, and so their product decreases exponentially quickly. In Claim 6.4, we show that the number of items capped in every round is also bounded. Taken together, these claims imply that the total error is small (see the discussion closing the utility analysis for the exact parameters).

Claim 6.2 (Error bound as a function of capped and Z_T)**.** *After round T of boosting, let $U_T \subseteq [m]$ and Z_T be as defined above. The total (fractional) error of the hypothesis H is at most $\prod_{t=1}^{T} Z_t + (1 - |U_T|/m)$.*

Proof. Examine the distribution D_{T+1}. We know that for every $i \in U_T$ (i.e. items never capped):

$$D_{T+1}[i] \quad = \quad (1/m) \cdot \prod_{t=1}^{T} \frac{e^{-\alpha \cdot c_{i,t}}}{Z_t} = \frac{e^{-\sum_t \alpha \cdot c_{i,t}}}{m \prod_t Z_t}$$

For the final hypothesis $H = \text{majority}\{h_1, \ldots, h_T\}$, we have:

$$[[H(x_i) \neq y_i]] \leq [[\sum_{t=1}^{T} c_{i,t} \geq 0]] \leq e^{-\sum_{t=1}^{T} \alpha \cdot c_{i,t}}$$

We get that:

$$\begin{aligned} \frac{1}{m} \sum_{i=1}^{m} [[H(x_i) \neq y_i]] &\leq \left(1 - \frac{|U_t|}{m}\right) + \frac{1}{m} \cdot \sum_{i \in U_T} e^{-\sum_t \alpha \cdot c_{i,t}} \\ &= \left(1 - \frac{|U_t|}{m}\right) + \sum_{i \in U_T} \prod_t Z_t \cdot D_{T+1}[i] \\ &\leq \left(1 - \frac{|U_t|}{m}\right) + \prod_t Z_t \end{aligned}$$

□

Claim 6.3 ([31], upper bound for Z_t)**.** *In every round t of boosting, $Z_t \leq \sqrt{1 - 4\eta^2}$, where η is the guaranteed advantage of the base learner.*

Proof. We have

$$\begin{aligned} Z_t &= \sum_{i=1}^{m} D_t[i] \cdot r_{i,t} \\ &\leq \sum_{i=1}^{m} D_t[i] \cdot u_{i,t} \\ &= \sum_{i=1}^{m} D_t[i] \cdot e^{-\alpha \cdot c_{i,t}} \\ &= \sum_{i=1}^{m} D_t[i] \left(\frac{1 + c_{i,t}}{2} \cdot e^{-\alpha} + \frac{1 - c_{i,t}}{2} \cdot e^{\alpha}\right) \\ &= \left(\frac{1 + \eta_t}{2}\right) \cdot e^{-\alpha} + \left(\frac{1 - \eta_t}{2}\right) \cdot e^{\alpha} \\ &\leq \left(\frac{1 + \eta}{2}\right) \cdot e^{-\alpha} + \left(\frac{1 - \eta}{2}\right) \cdot e^{\alpha} \end{aligned}$$

to minimize this bound on Z_t, we choose

$$\alpha = 1/2 \ln(\frac{1 + 2\eta}{1 - 2\eta})$$

where $\eta_t = (\sum_i D_t[i] \cdot c_{i,t})/2 \geq \eta$ is the advantage (over 1/2) of the base learner. This gives:

$$Z_t \leq \left(\sqrt{1-4\eta^2}\right) \cdot \sum_i D_t[i] = \sqrt{1-4\eta^2}$$

□

Claim 6.4 (Bound on number of items capped)**.** *In each round t, the number of items whose weight is capped is at most $\widehat{\ell_t}$.*

Proof. We choose w_t so that $\widehat{\ell_t} = \sum_{i=1}^{m} \min(v_{i,t-1} \cdot \exp(-\alpha \cdot c_{i,t})/w_t, 1)$. Each item whose weight is capped contributes 1 to the right-hand sum in this equality. Since the entire sum equals $\widehat{\ell_t}$, there can be at most that many items whose weight is capped. □

Now to argue utility (assuming that all noise values are small and the base learner always succeeds, as above), observe that whenever the boosting procedure halts before round T, the output is indeed a low-error hypothesis. From Claim 6.2, together with Claims 6.3 and 6.4, we conclude that the total error after T rounds of boosting is at most

$$\left(\sqrt{1-4\eta^2}\right)^T + T \cdot 3\ell/m$$

For appropriate choice of $T = O((1/\eta^2) \cdot \log(1/\gamma))$ we get that the error is at most

$$\gamma/4 + O((1/\eta^2) \cdot \log(1/\gamma) \cdot \ell/m)$$

it suffices to take

$$m = O((1/\eta^2) \cdot \ell \cdot \log(1/\gamma) \cdot (1/\gamma))$$

and we get that the total error is bounded by γ. With all but $(T \cdot \beta + \exp(-\Omega(\varepsilon \cdot \ell)))$ probability, the boosting procedure will indeed output a γ-error hypothesis if it halts at the end of round T. We set ℓ large enough so the failure probability becomes $(T \cdot \beta + \exp(-\kappa))$ ($\ell = O(\kappa/\varepsilon) = O(-\kappa \cdot T/\varepsilon^*)$ suffices).

Privacy Analysis. Fix a view up to the t-th hypothesis, including the hypotheses $h_1, \ldots, h_t$, and smoothing thresholds $w_1, \ldots, w_{t-1}$. We will argue that given such a fixed view from past rounds, for any two adjacent databases DB and DB', differing only on element i, we can compute the new termination decision (in Step 3), the new smoothing threshold w_t, and the new hypothesis h_{t+1} for round t. The view is also differentially private (see below for the exact bound). By composition of differential privacy, this will imply that the boosted learner's output $h_1, \ldots, h_t$ is differentially private. We begin with a helpful auxiliary lemma, a generalization of similar statements about amplifying differential privacy by sampling that appeared in [20, 2]. We then proceed with the bound for the privacy loss in each round.

Lemma 6.5. *Let A and B be distributions on universe U with statistical difference at most σ, and $M : U^k \to V$ an (ε, δ)-differentially private mechanism, where $k > 0$ and $\varepsilon \leq 1/2$. Then:*

$$D_\infty^{\delta \cdot (1+e^{2\varepsilon \cdot \sigma \cdot k})}(M(A^k), M(B^k)) \leq 4\varepsilon \cdot \sigma \cdot k$$

Proof. Since the statistical distance between A and B is σ, we can decompose both distributions as $A = (1-\sigma)\cdot C + \sigma \cdot A'$ and $B = (1-\sigma)\cdot C + \sigma \cdot B'$ for some distributions C, A', B' over U. We consider the distributions A^k, B^k and C^k and show that A^k and C^k are $(2\varepsilon \cdot \sigma \cdot k, \delta)$-differentially private. A similar bound holds for B^k and C^k, and so we deduce that A^k and B^k are $(4\varepsilon \cdot \sigma \cdot k, \delta \cdot (1 + e^{2\varepsilon \cdot \sigma \cdot k}))$-differentially private.

We can model sampling from A^k and C^k (respectively) by first tossing k coins of bias σ, and then sampling according to C when the coin is 0 (w.p. $1-\sigma$), and sampling according to A' or C (respectively) when the coin is 1 (w.p. σ). For any possible output $W \subseteq V$, we can decompose its probability under $M(A^k)$ and $M(C^k)$ according to which samples were taken from C. Using $r \in \{0,1\}^k$ to denote the k coins chosen in the sampling procedure, and D_r to denote the distribution of samples over U^k where the i-th sample is drawn from C when $r_i = 0$ and from A otherwise, and let $|r|$ be the weight of r (the number of ones). We get:

$$\Pr[M(A^k) \in W] = \sum_{r \in \{0,1\}^k} (1-\sigma)^{k-|r|} \cdot \sigma^{|r|} \cdot \Pr[M(D_r) \in W]$$

$$\Pr[M(C^k) \in W] = \sum_{r \in \{0,1\}^k} (1-\sigma)^{k-|r|} \cdot \sigma^{|r|} \cdot \Pr[M(C^k) \in W]$$

By (ε, δ)-differential privacy of M, we know that for $W \subseteq V$ and $r \in \{0,1\}^r$ of weight $|r|$ (the number of non-zero entries), it is the case that

$$e^{-\varepsilon \cdot |r|} \cdot M(C^k)[W] - \delta \leq M(D_r)[W] \leq e^{\varepsilon \cdot |r|} \cdot M(C^k)[W] + \delta$$

plugging this in, we get:

$$\begin{aligned}
M(A^k)[W] &= \sum_{r \in \{0,1\}^k} (1-\sigma)^{k-|r|} \cdot \sigma^{|r|} \cdot M(D_r)[W] \\
&\leq \sum_{r \in \{0,1\}^k} (1-\sigma)^{k-|r|} \cdot \sigma^{|r|} \cdot (e^{\varepsilon \cdot |r|} \cdot M(C^k)[W] + \delta) \\
&= M(C^k)[W] \cdot \left(\sum_{r \in \{0,1\}^k} (1-\sigma)^{k-|r|} \cdot \sigma^{|r|} \cdot e^{\varepsilon \cdot |r|} \right) + \delta \\
&= M(C^k)[W] \cdot ((1-\sigma) + \sigma \cdot e^{\varepsilon})^k + \delta \\
&\leq M(C^k)[W] \cdot (1 + 2\varepsilon \cdot \sigma)^k + \delta \\
&\leq M(C^k)[W] \cdot e^{2\varepsilon \cdot \sigma \cdot k} + \delta
\end{aligned}$$

where we used the fact that for $\varepsilon \leq 1/2$, we have $e^{\varepsilon} \leq 1 + 2\varepsilon$. Similarly, we get that:

$$M(A^k)[W] \geq M(C^k)[W] \cdot e^{-2\varepsilon \cdot \sigma \cdot k} - \delta$$

Similar claims hold for B^k, and so we get that A^k and B^k are $(4\varepsilon \cdot \sigma \cdot k, 2\delta)$-differentially private. □

Now to argue privacy for the people-boosting algorithm, we again view the accesses to the database as a composition of (adaptively chosen) queries on the database. We examine round t of boosting, starting at Step 3, as activating a mechanism M_t on the information computed

in the previous rounds and the (sensitive) database. The information computed in round t is the termination decision d_t and the smoothing threshold w_t, and then (in the next round) the hypothesis h_{t+1}. We denote the mechanism's operation in round t as:

$$(d_t, w_t, h_{t+1}) \leftarrow \mathcal{M}_t(x, h_1, d_1, w_1, h_2 \ldots, d_{t-1}, w_{t-1}, h_t)$$

Observe that the given history is sufficient to compute the output for round t: we can compute the $c_{i,j}$ and $v_{i,j}$ values for all $i \in [m]$ and rounds $j < t$. For round t we can then compute the termination check in Step 3 and the $c_{i,t}$ values. In turn, this allows us to compute the smoothing threshold w_t and from it the $v_{i,t}$ values, the distribution D_{t+1}, the next sampled set C_{t+1} and h_{t+1}. The following lemma shows that for any fixed history the mechanism M_t activated on that history and the (sensitive) database is differentially private.

Lemma 6.6. *For $t \geq 0$ fix a prior history $(h_1, d_1, w_1, h_2 \ldots, d_{t-1}, w_{t-1}, h_t)$, then the mechanism $\mathcal{M}_t(x, h_1, d_1, w_1, h_2 \ldots, d_{t-1}, w_{t-1}, h_t)$ (where all the inputs but x are fixed) is $(12\varepsilon_{base} \cdot k/\ell, \delta_{base} \cdot (1 + e^{6\varepsilon_{base} \cdot k/\ell}))$-differentially private.*

Proof. We analyze the algorithm's privacy guarantees step by step:

1. The sensitivity of the sum computed in Step 3 is only 1. By adding noise $\mathrm{Lap}(1/\varepsilon)$ we are assured that the termination decision is ε-differentially private.

2. For the smoothing threshold w_t, fix some possible value w and examine its probability by DB and DB'. For DB, we get that $w_t = w$ if $\widehat{\ell_t} = \sum_{i=1}^{m} \min(u_{i,t}/w, 1)$. The sensitivity of this sum on the right-hand side is at most 1, and so for DB' $w_t = w$ if $\widehat{\ell_t}$ is equal to a value at distance at most 1 from $\sum_{i=1}^{m} \min(u_{i,t}/w, 1)$. Since $\widehat{\ell_t}$ is chosen by adding Laplace noise of magnitude $(1/\varepsilon)$ to ℓ, we conclude that the probabilities of $w_t = w$ by DB and DB' differ by an e^{ε} multiplicative factor. I.e, w_t is also ε-differentially private.

3. Finally, we argue that (assuming the algorithm didn't terminate), the distribution D_{t+1} and D'_{t+1} obtained via DB and DB' are within statistical distance $3/\ell$ of each other. By Lemma 6.5, this implies that the hypothesis h_{t+1} is also $(12\varepsilon_{base} \cdot k/\ell, \delta_{base} \cdot (1 + e^{6\varepsilon_{base} \cdot k/\ell}))$-differentially private.

   To argue about the statistical distance, fix w_t as above, and i be the item on which the two adjacent datasets differ. We know that the noisy $\widehat{\ell_t}$ and $\widehat{\ell_t}'$ differ by at most 1, and that both are larger than $\ell - 1$ (otherwise the algorithm would have terminated).

   For all $j \neq i$ we have $c_{j,t} = c'_{j,t}$, and so (since we have fixed w_t) we have $v_{j,t} = v'_{j,t}$. For all $j \in [m]$ we know that $D_{t+1}[j] = v_{j,t}/\widehat{\ell_t}$ and $D'_{t+1}[j] = v_{j,t}/\widehat{\ell_t}'$. Now if $c_{j,t} = 1$ and $c'_{j,t} = -1$, then $\widehat{\ell_t} \geq \widehat{\ell_t}'$, and the statistical difference is at most

$$(v_{i,t}/\widehat{\ell_t} - v'_{i,t}/\widehat{\ell_t}') \leq (e^{\alpha} - e^{-\alpha})/\widehat{\ell_t}' \leq 3/\ell$$

   where here we use the fact that statistical distance between D_{t+1} and D'_{t+1} is (by definition) the sum of probability differences over items on which D_{t+1} has higher probability than D'_{t+1} (and the bounds $\alpha \leq 2\eta$, $\eta \leq 1/2$). Similarly, if $c_{j,t} = -1$ and $c'_{j,t} = 1$ then the statistical distance is at most

$$(v'_{i,t}/\widehat{\ell_t}' - v_{i,t}/\widehat{\ell_t}) \leq (e^{\alpha} - e^{-\alpha})/\widehat{\ell_t} \leq 3/\ell$$

By composition of differential privacy (Theorem 3.1), we conclude that M_t is (in total)$(12\varepsilon_{base} \cdot k/\ell, \delta_{base} \cdot (1 + e^{6\varepsilon_{base} \cdot k/\ell}))$-differentially private.

□

By composition of differential privacy (Theorem 3.1) over the T rounds of boosting, we conclude that the boosting for people algorithm in its entirety is $(2\varepsilon T + 12\varepsilon_{base} \cdot T \cdot k/\ell, T \cdot \delta_{base} \cdot (1 + e^{6\varepsilon_{base} \cdot k/\ell}))$-differentially private. We take $\varepsilon = \varepsilon^*/2T$, $\ell \geq 12Tk$, and get that the output is $(\varepsilon^* + \varepsilon_{base}, T \cdot \delta_{base} \cdot (1 + e^{\varepsilon_{base}}))$-differentially private.[2]

□

References

[1] B. Barak, M. Hardt, and S. Kale. The uniform hardcore lemma via approximate bregman projections. In *Proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2009.

[2] A. Beimel, S. P. Kasiviswanathan, and K. Nissim. Bounds on the sample complexity for private learning and private data release. In *TCC*, pages 437–454, 2010.

[3] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In *CRYPTO*, pages 26–45, 1998.

[4] A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: The SuLQ framework. In *Proceedings of the 24th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, June 2005.

[5] A. Blum, K. Ligett, and A. Roth. A learning theory approach to non-interactive database privacy. In *Proceedings of the 40th ACM SIGACT Symposium on Thoery of Computing*, 2008.

[6] K. Chaudhuri and C. Monteleoni. Privacy-preserving logistic regression. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2008.

[7] T. M. Cover and J. A. Thomas. *Elements of information theory.* Wiley-Interscience [John Wiley & Sons], Hoboken, NJ, second edition, 2006.

[8] I. Dinur and K. Nissim. Revealing information while preserving privacy. In *Proceedings of the Twenty-Second ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 202–210, 2003.

[9] C. Dwork. A firm foundation for private data analysis. *Communications of the ACM (to appear).*

[10] C. Dwork. An ad omnia approach to defining and achieving private data analysis. In F. Bonchi, E. Ferrari, B. Malin, and Y. Saygin, editors, *Privacy, Security, and Trust in KDD, First ACM SIGKDD International (PinKDD), Revised Selected Papers*, volume 4890 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 2007.

[2] Note we could also use here (and in Lemma 6.6) the improved composition from Theorem 3.3 to obtain better parameters with some degradation in the δ term of the privacy guarantee.

[11] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: privacy via distributed noise generation. In *Advances in Cryptology: Proceedings of EUROCRYPT*, pages 486–503, 2006.

[12] C. Dwork and J. Lei. Differential privacy and robust statistics. In *Proceedings of the 2009 International ACM Symposium on Theory of Computing (STOC)*, 2009.

[13] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Theory of Cryptography Conference*, pages 265–284, 2006.

[14] C. Dwork, M. Naor, O. Reingold, G. N. Rothblum, and S. Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In *STOC*, pages 381–390, 2009.

[15] C. Dwork and K. Nissim. Privacy-preserving datamining on vertically partitioned databases. In *Proceedings of CRYPTO 2004*, volume 3152, pages 528–544, 2004.

[16] Y. Freund. An improved boosting algorithm and its implications on learning complexity. In *The Fifth Annual Conference on Learning Theory (COLT)*, pages 391–398, 1992.

[17] D. Gavinsky. Optimally-smooth adaptive boosting and application to agnostic learning. *Journal of Machine Learning Research*, 4.

[18] B. Green and T. Tao. The primes contain arbitrarily long arithmetic progressions. *Annals of Mathematics*, 167:481–547, 2008.

[19] M. Hardt and G. N. Rothblum. A multiplicative weights mechanism for interactive privacy-preserving data analysis. *FOCS (to appear)*, 2010.

[20] S. Kasiviswanathan, H. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. What can we learn privately? In *Proceedings of FOCS 2008*, 2008.

[21] A. Klivans and R. Servedio. Boosting and hard-core set construction. *Machine Learning*.

[22] A. Machanavajjhala, D. Kifer, J. Abowd, J. Gehrke, and L. Vihuber. Privacy: From theory to practice on the map. In *Proc. ICDE*, 2008.

[23] F. McSherry. Privacy integrated queries (codebase). available on Microsoft Research downloads website. See also the Proceedings of SIGMOD 2009.

[24] F. McSherry. Privacy integrated queries. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 2009.

[25] F. McSherry and O. Williams. Probabilistic inference and differential privacy, 2009. Manuscript.

[26] I. Mironov. Personal communication, 2009.

[27] I. Mironov, O. Pandey, O. Reingold, and S. Vadhan. Computational differential privacy. In S. Halevi, editor, *Advances in Cryptology—CRYPTO '09*, volume 5677 of *Lecture Notes in Computer Science*, pages 126–142. Springer-Verlag, 16–20 August 2009.

[28] O. Reingold, L. Trevisan, M. Tulsiani, and S. Vadhan. Dense subsets of pseudorandom sets. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS '08)*, pages 76–85. IEEE, 26–28 October 2008.

[29] B. Rubinstein, P. Bartlett, L. Huang, and N. Taft. Learning in a large function space: Privacy-preserving mechanisms for svm learning, 2009. http://arxiv.org/abs/0911.5708.

[30] R. Schapire. The boosting approach to machine learning: An overview. In *D. D. Denison, M. H. Hansen, C. Holmes, B. Mallick, B. Yu, editors, Nonlinear Estimation and Classification.* Springer, 2003.

[31] R. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 39:297–336, 1999.

[32] R. Servedio. Smooth boosting and learning with malicious noise. *Journal of Machine Learning Research.*

[33] T. Tao and T. Ziegler. The primes contain arbitrarily long polynomial progressions. *Acta Mathematica*, 201:213–305, 2008.

[34] J. Ullman and S. Vadhan. PCPs and the hardness of generating synthetic data. Technical Report TR10-017, Electronic Colloquium on Computational Complexity, February 2010.

A Additional Related Work

The literature contains several results on privacy-preserving learning. In [4] it is shown that anything computable in the statistical queries learning model can be computed with privacy at little cost in accuracy. Similarly, [20] show that everything PAC learnable can be learned in a differentially private fashion using polynomially many samples (but not necessarily in polynomial time), again with good accuracy. Privacy-preserving support vector machines are studied in [29].

Chaudhuri and Monteleoni investigated privacy-preserving logistic regression [6], introducing a beautiful general new technique for optimization problems. Succinctly put, their idea is to perturb the optimization function rather than the solution. McSherry and Williams show how one can reduce errors after programming with a differentially private programming interface (as first done in [4], extended in [24], and built in [23]) by applying probabilistic inference techniques to the (privacy-preserving) intermediate results [25].

The vast literature on boosting continues to grow. An excellent summary of roughly the first decade of boosting can be found in [30], and the starting point for both our algorithms is a variant of AdaBoost due to Schapire and Singer [31]. Our Boosting for People algorithm yields is a *smooth* boosting algorithm [32, 21, 17, 1]. This means that no single example is assigned too much weight. This is natural from a privacy perspective, since we require the behavior of the algorithm to be "almost the same" on databases differing in the presence or absence of a single individual. It also raises the possibility that adapting an existing or future smooth boosting algorithm to preserve privacy might yield a simpler algorithm.

Our Boosting for Queries algorithm must cope with real-valued labels. This was first handled in [16]. This algorithm also introduces some noise in labeling queries, suggesting a possible connection to boosting with noise, again rasing the possibility that adapting an existing or future noisy boosting algorithm to preserve privacy might yield an improvement over our algorithm.

Two recent works give *interactive* mechanisms for answering a set $\mathcal{Q}$ of counting queries, where the queries do not have to be specified in advance (the results above are all for query sets specified in advance). Roth and Roughgarden give a mechanism where the error's dependence on n is polynomial (roughly $n^{2/3}$, as in [5]), and the dependence on $|\mathcal{Q}|$ is roughly poly-logarithmic. The running time is either super-polynomial in $|X|, |\mathcal{Q}|$, or can be made polynomial if accuracy is relaxed to hold only for all but a negligible fraction of databases (which depend on the queries). In subsequent work, Hardt and Rothblum [19] give a mechanism where the error's dependence on n is roughly $n^{1/2}$ (as in our work), and its dependence on $|\mathcal{Q}|$ is roughly logarithmic. The running time is either linear in $|X|$ and $|\mathcal{Q}|$, or can be made poly-logarithmic in $|X|$ if accuracy is relaxed to hold only for *smooth* databases (in particular, all but a negligible fraction of databases are smooth).

B Composition, Continued

This appendix contains details omitted from Section 3.

Proof of Lemma 3.2. We know that for any Y and Z it is the case that $D(Y||Z) \geq 0$ (via the "log-sum inequality," cf. [7]), and so it suffices to bound $D(Y||Z) + D(Z||Y)$. We get:

$$\begin{aligned}
D(Y||Z) &\leq D(Y||Z) + D(Z||Y) \\
&= \sum_y \left[\Pr[Y=y] \cdot \left(\ln \frac{\Pr[Y=y]}{\Pr[Z=y]} + \ln \frac{\Pr[Z=y]}{\Pr[Y=y]} \right) + (\Pr[Z=y] - \Pr[Y=y]) \cdot \left(\ln \frac{\Pr[Z=y]}{\Pr[Y=y]} \right) \right] \\
&\leq \sum_y [0 + |\Pr[Z=y] - \Pr[Y=y]| \cdot \varepsilon] \\
&= \varepsilon \cdot \sum_y [\max\{\Pr[Y=y], \Pr[Z=y]\} - \min\{\Pr[Y=y], \Pr[Z=y]\}] \\
&\leq \varepsilon \cdot \sum_y [(e^\varepsilon - 1) \cdot \min\{\Pr[Y=y], \Pr[Z=y]\}] \\
&\leq \varepsilon \cdot (e^\varepsilon - 1).
\end{aligned}$$

□

Lemma B.1 (Azuma's Inequality)**.** *Let $C_1, \ldots, C_k$ be real-valued random variables such that for every $i \in [k]$,*

1. $\Pr[|C_i| \leq \alpha] = 1$*, and*
2. *for every $(c_1, \ldots, c_{i-1}) \in \mathrm{Supp}(C_1, \ldots, C_{i-1})$, we have*

$$\mathrm{E}[C_i | C_1 = c_1, \ldots, C_{i-1} = c_{i-1}] \leq \beta.$$

Then for every $z > 0$, we have

$$\Pr\left[\sum_{i=1}^{k} C_i > k\beta + z\sqrt{k}\cdot\alpha\right] \leq e^{-z^2/2}.$$

Theorem B.2 (Theorem 3.3, restated)**.** *For every $\varepsilon > 0$, $\delta, \delta' \in [0,1]$, and $k \in \mathcal{N}$, the class of (ε,δ)-differentially private mechanisms is $(\varepsilon', k\delta + \delta')$-differentially private under k-fold adaptive composition, for*

$$\varepsilon' = \sqrt{2k\ln(1/\delta')}\cdot\varepsilon + k\cdot\varepsilon\varepsilon_0,$$

where $\varepsilon_0 = e^{\varepsilon} - 1$.

Proof of Theorem 3.3 for the case of $\delta = 0$. A view of the adversary A consists of a tuple of the form $v = (r, y_1, \ldots, y_k)$, where r is the coin tosses of A and $y_1, \ldots, y_k$ are the outputs of the mechanisms $\mathcal{M}_1, \ldots, \mathcal{M}_k$. Let

$$B = \{v : \Pr[V^0 = v] > e^{\varepsilon'}\cdot\Pr[V^1 = v]\}.$$

We will show that $\Pr[V^0 \in B] \leq \delta'$, and hence for every set S, we have

$$\Pr[V^0 \in S] \leq \Pr[V^0 \in B] + \Pr[V^0 \in (S \setminus B)] \leq \delta' + e^{\varepsilon'}\cdot\Pr[V^1 \in S].$$

This is equivalent to saying that $D_\infty^{\delta'}(V^0||V^1) \leq \varepsilon'$.

It remains to show $\Pr[V^0 \in B] \leq \delta'$. Let random variable $V^0 = (R^0, Y_1^0, \ldots, Y_k^0)$ denote the view of A in Experiment 0 and $V^1 = (R^1, Y_1^1, \ldots, Y_k^1)$ the view of A in Experiment 1. Then for a fixed view $v = (r, y_1, \ldots, y_k)$, we have

$$\begin{aligned}
\ln\left(\frac{\Pr[V^0 = v]}{\Pr[V^1 = v]}\right) &= \ln\left(\frac{\Pr[R^0 = r]}{\Pr[R^1 = r]}\cdot\prod_{i=1}^{k}\frac{\Pr[Y_i^0 = y_i | R^0 = r, Y_1^0 = y_1, \ldots, Y_{i-1}^0 = y_{i-1}]}{\Pr[Y_i^1 = y_i | R^1 = r, Y_1^1 = y_1, \ldots, Y_{i-1}^1 = y_{i-1}]}\right) \\
&= \sum_{i=1}^{k}\ln\left(\frac{\Pr[Y_i^0 = y_i | R^0 = r, Y_1^0 = y_1, \ldots, Y_{i-1}^0 = y_{i-1}]}{\Pr[Y_i^1 = y_i | R^1 = r, Y_1^1 = y_1, \ldots, Y_{i-1}^1 = y_{i-1}]}\right) \\
&\stackrel{\text{def}}{=} \sum_{i=1}^{k} c_i(r, y_1, \ldots, y_i).
\end{aligned}$$

Now for every prefix $(r, y_1, \ldots, y_{i-1})$ we condition on $R^0 = r, Y_1^0 = y_1, \ldots, Y_{i-1}^0 = y_{i-1}$, and analyze the expectation and maximum possible value of the random variable $c_i(R^0, Y_1^0, \ldots, Y_i^0) = c_i(r, y_1, \ldots, y_{i-1}, Y_i^0)$, . Once the prefix is fixed, the next pair of databases x_i^0 and x_i^1, the mechanism $\mathcal{M}_i$, and parameter w_i output by A are also determined (in both Experiment 0 and 1). Thus Y_i^0 is distributed according to $\mathcal{M}_i(w_i, x_i^0)$. Moreover for any value y_i, we have

$$c_i(r, y_1, \ldots, y_{i-1}, y_i) = \ln\left(\frac{\Pr[\mathcal{M}_i(w_i, x_i^0) = y_i]}{\Pr[\mathcal{M}_i(w_i, x_i^1) = y_i]}\right).$$

By ε-differential privacy this is bounded by ε. We can also reason as follows:

$$|c_i(r, y_1, \ldots, y_{i-1}, y_i)| \leq \max\left\{D_\infty(\mathcal{M}_i(w_i, x_i^0)||\mathcal{M}_i(w_i, x_i^1)), D_\infty(\mathcal{M}_i(w_i, x_i^1)||\mathcal{M}_i(w_i, x_i^0))\right\} = \varepsilon.$$

By Lemma 3.2, we have:

$$\mathrm{E}\left[c_i(R^0, Y_1^0, \ldots, Y_i^0) | R^0 = r, Y_1^0 = y_1, \ldots, Y_{i-1}^0 = y_{i-1}\right] = D(\mathcal{M}_i(w_i, x_i^0) || \mathcal{M}_i(w_i, x_i^1)) \leq \varepsilon \cdot \varepsilon_0.$$

Thus we can apply Azuma's Inequality to the random variables $C_i = c_i(R^0, Y_1^0, \ldots, Y_i^0)$ with $\alpha = \varepsilon$, $\beta = \varepsilon \cdot \varepsilon_0$, and $z = \sqrt{2\ln(1/\delta')}$, to deduce that

$$\Pr[V^0 \in B] = \Pr\left[\sum_i C_i > \varepsilon'\right] < e^{-z^2/2} = \delta',$$

as desired. □

Proof of Theorem 3.3 for the case of $\delta > 0$. We use the characterization of approximate max-divergence from Lemma 2.1 (Part 1) to reduce the analysis to the same situation as in the case of ε-differential privacy as follows.

We use the same notation $V^b = (R^b, Y_1^b, \ldots, Y_k^b)$ as in the above proof of Theorem 3.3 for the case where $\delta = 0$. Using Lemma 2.1, Part 1 for each of the differentially private mechanisms selected by the adversary A and the triangle inequality for statistical distance, it follows that that V^0 is $k\delta$-close to a random variable $W = (R, Z_1, \ldots, Z_k)$ such that for every prefix $r, y_1, \ldots, y_{i-1}$, if we condition on $R = R^1 = r, Z_1 = Y_1^1 = y_1, \ldots, Z_{i-1} = Y_{i-1}^1 = y_{i-1}$, then it holds that $D_\infty(Z_i || Y_i^1) \leq \varepsilon$ and $D_\infty(Y_i^1 || Z_i) \leq \varepsilon$.

This suffices for the argument in the above proof for the case $\delta = 0$ to show that $D_\infty^{\delta'}(W || V^1) \leq \varepsilon'$. Since V^0 is $k\delta$-close to W, Lemma 2.1 gives $D^{\delta' + k\delta}(V^0 || W) \leq \varepsilon'$. □