Journal of Privacy and Confidentiality
Vol. 13 (1) 2023

TPDP 2021

Submitted Dec 2021
Published Aug 2023

PRIVATE BOOSTED DECISION TREES VIA SMOOTH RE-WEIGHTING

VAHID R. ASADI, MARCO L. CARMOSINO, MOHAMMADMAHDI JAHANARA, AKBAR RAFIEY*, AND BAHAR SALAMATIAN

Department of Computer Science, Simon Fraser University, Burnaby, BC, Canada
e-mail address: vasadi@sfu.ca

Department of Computer Science, Boston University, Boston, MA, United States
e-mail address: marco@ntime.org

Scroll Foundation, Seychelles
e-mail address: mohammad@scroll.io

HDSI, University of California, San Diego, CA, United States
e-mail address: arafiey@ucsd.edu

Department of Computer Science, Simon Fraser University, Burnaby, BC, Canada
e-mail address: bsalamat@sfu.ca

Abstract. Protecting the privacy of people whose data is used by machine learning algorithms is important. Differential Privacy is the appropriate mathematical framework for formal guarantees of privacy, and boosted decision trees are a popular machine learning technique. We propose and test a practical algorithm for boosting decision trees that guarantees differential privacy. Privacy is enforced because our booster never puts too much weight on any one example; this ensures that each individual's data never influences a single tree "too much." Experiments show that this boosting algorithm can produce better model sparsity and accuracy than other differentially private ensemble classifiers.

1. Introduction

Boosted decision trees are a popular, widely deployed, and successful machine learning technique. Boosting constructs an ensemble of decision trees sequentially, by calling a decision tree *base learner* with sample weights that "concentrate attention" on training examples that are poorly classified by trees constructed so far (Schapire and Freund, 2012).

Differential Privacy (DP) is a mathematical definition of privacy which ensures that the distribution over hypotheses produced by a learning algorithm does not depend "too much" (quantified by ϵ) on any one input example (Dwork, 2006). An adversary cannot even tell if

Key words and phrases: Differentially Private Boosting, Decision Trees, Smooth Boosting.
Authors in alphabetic order.
*Corresponding author.

www.journalprivacyconfidentiality.org
DOI:10.29012/jpc.808

a specific individual participated in a differentially private study or not (see Wood et al., 2018, section IV.C.1).

Recent purely theoretical work used *Smooth Boosting* — algorithms that never concentrate too much sample weight on any one example — to give a simple and differentially private algorithm for learning large-margin half-spaces (Bun et al., 2020). Their boosting algorithm is generic; it does not depend on any specific features of the weak learner beyond differential privacy.

1.1. **Our Contributions.** Here, we demonstrate that the smooth boosting algorithm of Bun et al. (2020) is a practical and efficient differentially private classifier when paired with decision "stumps" — depth-1 trees. We compare on three classification tasks to DP logistic regression (Chaudhuri et al., 2011), DP bagging (Jordon et al., 2019), DP gradient boosting (Li et al., 2020), and smooth boosting over our own "reference implementation" of DP decision trees. In all cases, smooth-boosted decision *stumps* improve on other algorithms in accuracy, model sparsity, or both in the high-privacy regime. This is surprising; in the non-private setting somewhat deeper trees (depth 3 - 7) generally improve accuracy. It seems that stumps better tolerate the amount of noise that must be added to enforce privacy for small samples. Since many applications of DP (e.g., US Census sample surveys, genetic data) require simple and accurate models for small datasets, we regard the high utility of smooth-boosted DP-Stumps in these settings as a *pleasant* surprise. In order to analyze the privacy of our algorithm, we introduce the notion of Weighted Exponential Mechanism and Weighted Return Noisy Max Mechanism based on the novel notion of robust sensitivity which we believe is of independent interest.

1.2. **Related Work.** Decision trees are one of the most popular classifiers; often used for their efficiency and interpretability. Since the NP-completeness result of Hyafil and Rivest (1976), there has been an extensive body of research devoted to designing heuristic algorithms for inducing decision trees. These algorithms are efficient and successful in practice (Rokach and Maimon, 2014). Notable examples are greedy procedures such as ID3, C4.5, and CART (Quinlan, 1986, 1993; Breiman et al., 1984). They iteratively "grow" a single tree by adding children to some leaf node of an existing tree according to a *splitting criterion*.

Differentially Private Decision Trees. Many previous works explored differentially private algorithms for learning *single* decision trees. Authors in Blum et al. (2005) showed how a traditional non-private algorithm (ID3) could be modified to achieve differential privacy by adding noise to the splitting criterion. Friedman and Schuster (2010) empirically demonstrated the effectiveness of using the exponential mechanism to privately select splits for ID3 and C4.5.

Recent work modified the TopDown algorithm of Kearns and Mansour (1996) to enforce differential privacy (Wang et al., 2020). This is particularly interesting because TopDown is *not* a heuristic. Under a *weak learning* assumption — if the features considered for splitting have some advantage over random guessing — TopDown is guaranteed to learn a tree with low training error. Wang et al. (2020) preserve this guarantee under differential privacy by appealing to the utility of the Laplace Mechanism. Here, we implement a simpler DP-TopDown algorithm — as the goal of our work is to test differentially private *boosting,* weaker tree induction is perferable.

Differentially Private Boosting. Differentially private boosting is less well-studied because the iterative structure of boosting algorithms complicates the task of enforcing privacy while maintaining utility. In theory, Dwork et al. (2010) designed the first differentially private boosting algorithm. Later, Bun et al. (2020) offered a much simpler private algorithm based on the hard-core lemma of Barak et al. (2009). Both algorithms preserved privacy by using "smooth" distributions over the sample to limit the "attention" any one example receives from a base learner. Our `LazyBB` (Algorithm 1) is an implementation of the algorithm of Bun et al. (2020) over decision trees and stumps.

Boosting by reweighting updates an explicit distributions over the data, where the probability mass on an example reflects how difficult it is to classify. *Gradient Boosting* iteratively fits the residuals of the combined voting classifier — it alters the labels instead of explicit weights on each sample.

One very recent experimental work studies differentially private *gradient* tree boosting (Li et al., 2020). Their base learner is an ensemble of greedily-constructed decision trees on disjoint subsets of the data, so that parallel composition may be used *inside* the base learner to save privacy. They deal with the "too much attention" problem by *clipping* the pseudo-residuals at each round, so that outliers do not compromise privacy by over-influencing the hypothesis at any round. They use composition to spread the privacy budget across each round of boosting.

Our algorithm is boosting by *reweighting* and uses much simpler base learners. Our update rule is just multiplicative weights, and we enforce privacy by *projecting* the resulting distribution over examples into the space of smooth distributions. Our algorithm remains accurate in the *high-privacy* ($\epsilon < 1$) setting; Li et al. (2020) did not explore this regime.

2. Preliminaries

2.1. Distributions and Smoothness.

To preserve privacy, we will never concentrate too much "attention" on a single example. This can be enforced by only using *smooth distributions* — where no example is allowed to have too much relative weight.

Definition 2.1 (κ-Smooth Distributions)**.** A probability distribution μ on domain X is *κ-smooth* if for each $x \in X$ we have $\mu(x) \leq \frac{1}{\kappa|X|}$, where $\kappa \in [0, 1]$.

To maintain the invariant that we only call base learners on smooth distributions, we Bregman-project onto the space of *high density measures*. High density measures[1] correspond to smooth probability distributions. Indeed, the measure μ over X has density at least κ if and only if the probability distribution $\frac{1}{|\mu|}\mu$ satisfies smoothness $\mu(x) \leq \frac{|\mu|}{\kappa|X|}$ for all $x \in X$ where $|\mu| = \sum_{x \in X} \mu(x)$ and density of μ is $|\mu|/|X|$.

Definition 2.2 (Bregman Projection)**.** Let $\Gamma \subseteq \mathbb{R}^{|S|}$ be a non-empty closed convex set of measures over S. The *Bregman projection* of $\tilde{\mu}$ onto Γ is defined as: $\Pi_\Gamma \tilde{\mu} = \arg\min_{\mu \in \Gamma} \mathrm{KL}(\mu \parallel \tilde{\mu})$.

The result of Bregman (1967) says Bregman projections do not badly "distort" KL-divergence. Moreover, when Γ is the set of κ-dense measures we can compute $\Pi_\Gamma \tilde{\mu}$ for measure $\tilde{\mu}$ with $|\tilde{\mu}| < \kappa|X|$ (Barak et al., 2009). Finally, we require the following notion of similarity.

[1] A *measure* is a function from the domain to $[0, 1]$ that need not sum to one; normalizing measures to total weight naturally results in a distribution.

Definition 2.3 (Statistical Distance)**.** The *statistical distance*, a.k.a. *total variation distance*, between two distributions μ and ν on Ω, denoted $\mathtt{d}(\mu, \nu)$, is defined as $\mathtt{d}(\mu, \nu) = \max_{S \subset \Omega} |\mu(S) - \nu(S)|$.

For finite sets Ω, $\mathtt{d}(\mu, \nu) = 1/2 \sum_{x \in \Omega} |\mu(x) - \nu(x)|$ e.g., see Proposition 4.2 in Levin and Peres (2017).

2.2. **Learning.** Throughout the paper we let $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}^n$ where $\mathbf{x}_i = (x_{i1}, \ldots, x_{ir})$ and $y_i \in \{+1, -1\}$ denote a dataset where *all* features and labels are Boolean. Though our techniques readily extend to continuous-feature or multi-label learning, studying this restricted classification setting simplifies the presentation and experiments for this short paper.

Definition 2.4 (Weak Learner)**.** Let $S \subset (\mathcal{X} \times \{\pm 1\})^n$ be a training set of size n. Let μ be a distribution over $[n]$. A *weak learning algorithm* with *advantage* γ takes (S, μ) as input and outputs a function $h : \mathcal{X} \to \{\pm 1\}$ such that: $\Pr_{x \sim \mu}[h(x) = c(x)] \geq 1/2 + \gamma$

Definition 2.5 (Margin)**.** For binary classification, the *margin* (denoted σ) of an ensemble $H = h_1, \ldots, h_\tau$ consisting of τ hypotheses on an example (x, y) is a number between $-\tau$ and τ that captures how "right" the classifier as a whole is $\sigma(H, x, y) = y \sum_{j=1}^{\tau} h(x)$.

2.3. **Differential Privacy.** The definition of differential privacy relies on the notion of neighboring datasets. We say two datasets are neighboring if they differ in a single record. We write $D \sim D'$ when two datasets D, D' are neighboring.

Definition 2.6 ((ϵ, δ)-Differential Privacy (Dwork et al., 2006))**.** For $\epsilon, \delta \in \mathbb{R}_+$, we say that a randomized computation M is (ϵ, δ)-*differentially private* if for any neighboring datasets $D \sim D'$, and for any set of outcomes $S \subseteq \text{range}(M)$,

$$\Pr[M(D) \in S] \leq \exp(\epsilon) \Pr[M(D') \in S] + \delta.$$

When $\delta = 0$, we say M is ϵ-*differentially private*.

Differentially private algorithms must be calibrated to the sensitivity of the function of interest with respect to small changes in the input dataset, defined formally as follows.

Definition 2.7 (Sensitivity)**.** The sensitivity of a function $F \colon X \to \mathbb{R}^k$ is $\max_{D \sim D' \in X} ||F(D) - F(D')||_1$. A function with sensitivity Δ is called Δ-sensitive.

Two privacy composition theorems, namely sequential composition and parallel composition, are widely used in the design of mechanisms.

Theorem 2.8 (Sequential Composition (Bun and Steinke, 2016; Dwork and Lei, 2009; Dwork et al., 2010; McSherry and Talwar, 2007))**.** *Suppose a set of privacy mechanisms* $M = \{M_1, \ldots, M_k\}$ *are sequentially performed on a dataset, and each* M_i *is* (ϵ_i, δ_i)*-differentially private with* $\epsilon_i \leq \epsilon_0$ *and* $\delta_i \leq \delta_0$ *for every* $1 \leq i \leq k$*. Then mechanism* M *satisfies* (ϵ, δ)*-differential privacy where*

- $\epsilon = k\epsilon_0$ *and* $\delta = k\delta_0$ *(the basic composition), or*
- $\epsilon = \sqrt{2k \ln 1/\delta'}\epsilon_0 + k\epsilon_0(\mathrm{e}^{\epsilon_0} - 1)$ *and* $\delta = \delta' + k\delta_0$ *for any* $\delta' > 0$ *(the advanced composition).*

Theorem 2.9 (Parallel Composition (McSherry, 2010))**.** *Let $D_1, \ldots, D_k$ be a partition of the input domain and suppose $M_1, \ldots, M_k$ are mechanisms so that M_i satisfies ϵ_i-differential privacy. Then the mechanism $M(S) = (M_1(S \cap D_1), \ldots, M_k(S \cap D_k))$ satisfies $(\max_i \epsilon_i)$-differential privacy.*

2.4. **Differentially Private Learning.** Given two neighboring datasets and *almost* the same distributions on them, privacy requires weak learners to output the same hypothesis with high probability. This idea was formalized for zero-concentrated differential privacy (zCDP) in Definition 18 of Bun et al. (2020). Below, we adapt it for the (ϵ, δ)-DP setting.

Definition 2.10 (DP Weak Learning)**.** A weak learning algorithm $\texttt{WkL} : S \times \mathcal{D}(S) \to \mathcal{H}$ is $(\epsilon, \delta, \zeta)$-differentially private if for all neighboring samples $S \sim S' \in (\mathcal{X}^n \times \{\pm 1\})$ and all $H \subseteq \mathcal{H}$, and any pair of distributions $\hat{\mu}, \hat{\mu}'$ on $[n]$ with $\mathtt{d}(\hat{\mu}, \hat{\mu}') < \zeta$, we have:

$$\Pr[\texttt{WKL}(S, \hat{\mu}) \in H] \leq \exp(\epsilon) \Pr[\texttt{WKL}(S', \hat{\mu}') \in H] + \delta.$$

Note that the notion of sensitivity for differentially private weak learners depends on the promised total variation distance ζ. Hence, differentially private weak learners must be calibrated to the sensitivity of the function of interest with respect to small changes in the distribution on the dataset. For this purpose, we introduce *robust sensitivity* below. There is no analog of robust sensitivity in the zCDP setting of Bun et al. (2020), because their private weak learner for halfspaces did not require it — they exploited inherent "compatibility" between Gaussian noise added to preserve privacy and the geometry of large-margin halfspaces. We do not have this luxury in the (ϵ, δ)-DP setting, and so must reason directly about how the accuracy of each potential weak learner changes with the distribution over examples.

Definition 2.11 (Robust Sensitivity)**.** The robust sensitivity of a function $F\colon (X, \mathsf{M}) \to \mathbb{R}^k$ where M is the set of all distributions on X is defined as

$$\max_{\substack{D \sim D' \in X \\ \hat{\mu}, \hat{\mu}' \in \mathsf{M} : \mathtt{d}(\hat{\mu}, \hat{\mu}') < \zeta}} ||F(D, \hat{\mu}(D)) - F(D', \hat{\mu}'(D'))||_1.$$

A function with robust sensitivity Δ_ζ is called Δ_ζ robustly sensitive.

The standard Exponential Mechanism (McSherry and Talwar, 2007) does not consider utility functions with an auxiliary weighting μ. But for weak learning we only demand privacy (close output distributions) when *both* the dataset and measures are "close." When both promises hold and μ is fixed, the Exponential Mechanism is indeed a differentially private weak learner; see the Appendix for a proof.

Definition 2.12 (Weighted Exponential Mechanism)**.** Let $\eta > 0$ and let $q_{D,\mu}\colon \mathcal{H} \to \mathbb{R}$ be a quality score. Then, the *Weighted Exponential Mechanism* $WEM(\eta, q_{D,\mu})$ outputs $h \in \mathcal{H}$ with probability proportional to $\exp(\eta \cdot q_{D,\mu}(h))$.

Similar to the Exponential Mechanism one can prove privacy and utility guarantee for the Weighted Exponential Mechanism.

Theorem 2.13. *Suppose the quality score $q_{D,\mu}\colon \mathcal{H} \to \mathbb{R}$ has robust sensitivity Δ_ζ. Then, $WEM(\eta, q_{D,\mu})$ is $(2\eta\Delta_\zeta, 0, \zeta)$-differentially private weak learner. Moreover, for every $\beta \in (0, 1)$, $WEM(\eta, q_{D,\mu})$ outputs $h \in \mathcal{H}$ so that*

$$\Pr\left[q_{D,\mu}(h) \geq \max_{h' \in \mathcal{H}} q_{D,\mu}(h') - \ln\left(|\mathcal{H}|/\beta\right)/\eta\right] \geq 1 - \beta.$$

Another differentially private mechanism that we use is Weighted Return Noisy Max (WRNM). Let $f_1, \ldots, f_k$ be k quality functions where each $f_i : S \times \mathcal{D}(S) \to \mathbb{R}$ maps datasets and distributions over them to real numbers. For a dataset S and distribution μ over S, WRNM adds independently generated Laplace noise $Lap(1/\eta)$ to each f_i and returns the index of the largest noisy function i.e. $i^* = \underset{i}{\text{argmax}}(f_i + Z_i)$ where each Z_i denotes a random variable drawn independently from the Laplace distribution with scale parameter $1/\eta$.

Theorem 2.14. *Suppose each f_i has robust sensitivity at most Δ_ζ. Then WRNM is a $(2\eta\Delta_\zeta, 0, \zeta)$-differentially private weak learner.*

3. Private Boosting

Our boosting algorithm, Algorithm 1, simply calculates the current margin of each example at each round, exponentially weights the sample accordingly, and then calls a private base learner with smoothed sample weights. The hypothesis returned by this base learner is added to the ensemble H, then the process repeats. Privacy follows from (advanced) composition and the definitions of differentially private weak learning. Utility (low training error) follows from regret bounds for lazy projected mirror descent and a reduction of boosting to zero-sum games. Theorem 3.1 formalizes these guarantees; for the proof, see Bun et al. (2020). Next, we discuss the role of each parameter.

Round Count τ. The number of base hypotheses. In the non-private setting, τ is like a regularization parameter — we increase it until just before overfitting is observed. In the private setting, there is an additional trade-off: more rounds *could* decrease training error until the amount of noise we must inject into the weak learner at each round (to preserve privacy) overwhelms progress.

Learning Rate λ. Exponential weighting is attenuated by a *learning rate* λ to ensure that weights do not shift too dramatically between calls to the base learner. λ appears negatively because the margin is negative when the ensemble is incorrect. Signs cancel to make the weight on an example *larger* when the committee is bad, as desired.

Smoothness κ. Base learners attempt to maximize their probability of correctness over each intermediate distribution. Suppose the t-th distribution was a point mass on example x_i — this would pose a serious threat to privacy, as hypothesis h_t would only contain information about individual x_i! We ensure this never happens by invoking the weak learner only over κ-smooth distributions: each example has probability mass "capped" at $\frac{1}{\kappa n}$. For larger samples, we have smaller mass caps, and so can inject less noise to enforce privacy. Note that by setting $\kappa = 1$, we force each intermediate distribution to be uniform, which would entirely negate the effects of boosting: reweighting would simply be impossible. Conversely, taking $\kappa \to 0$ will entirely remove the smoothness constraint.

Theorem 3.1 (Privacy & Utility of `LazyBB`)**.** *Let L be a $(\epsilon_b, \delta_b, (1/\kappa n))$-DP weak learner with advantage γ and failure probability β for concept class $\mathcal{H}$. Running `LazyBB` with L for $\tau \geq \frac{16 \log(1/\kappa)}{\gamma^2}$ rounds on a sample of size n with $\lambda = \gamma/4$ guarantees:*

Privacy: *`LazyBB` is (ϵ_A, δ_A)-DP, where $\epsilon_A = \sqrt{2\tau \cdot \ln(1/\delta')} \cdot \epsilon_b + \tau \cdot \epsilon_b \cdot (\exp(\epsilon_b) - 1)$ and $\delta_A = \tau \cdot \delta_b + \delta'$ for every $\delta' > 0$ (using advanced composition).*

Utility: *With all but $(\tau \cdot \beta)$ probability, H has at least γ-good normalized margin on a $(1 - \kappa)$ fraction of S i.e., $\Pr_{(x,y)\sim S}\left[y/\tau \sum_{j=1}^{\tau} h_j(x) \leq \gamma\right] \leq \kappa$.*

Algorithm 1: `LazyBB`: Weighted Lazy-Bregman Boosting

Parameters: $\kappa \in (0,1)$, desired training error; $\lambda \in (0,1)$, learning rate; $\tau \in \mathbb{N}$ number of rounds

Input: $S \in X^n$, the sample;

$H \leftarrow \emptyset$ and $\mu_1(i) \leftarrow \kappa \ \forall i \in [n]$ {Uniform bounded measure}

for $t = 1$ to τ **do**

$\hat{\mu}_t \leftarrow$ Normalize μ_t to a distribution {Obtaining a κ-smooth distribution}

$h_t \leftarrow \texttt{WkL}(S, \hat{\mu}_t)$

$H \leftarrow H \cup \{h_t\}$

$\sigma_t(i) \leftarrow y_i \sum_{j=1}^{t} h_j(x_i) \ \forall i \in [n]$ {Normalized score of current majority vote}

$\tilde{\mu}_{t+1}(i) \leftarrow \exp\left(-\lambda \sigma_t(i)\right) \kappa \ \forall i \in [n]$

$\mu_{t+1} \leftarrow \Pi_\Gamma(\tilde{\mu}_{t+1})$ {Bregman project to a κ-dense measure}

end for

Output: $\hat{f}(x) = \mathrm{Maj}_{h_j \in H}\left[h_j(x)\right]$

Weak Learner failure probability β is critical to admit because whatever "noise" process a DP weak learner uses to ensure privacy may ruin utility on some round. We must union bound over this event in the training error guarantee.

4. Concrete Private Boosting

Here we specify concrete weak learners and give privacy guarantees for `LazyBB` combined with these weak learners.

4.1. Baseline: 1-Rules.

To establish a baseline for performance of both private and non-private learning, we use the simplest possible hypothesis class: 1-Rules or "Decision Stumps" (Iba and Langley, 1992; Holte, 1993). In the Boolean feature and classification setting, these are just constants or signed literals (e.g. $-x_{17}$) over the data domain.

$$\mathsf{1R}(\mathcal{S}) = \{x_i\}_{i\in[d]} \cup \{-x_i\}_{i\in[d]} \cup \{+1, -1\} \quad \text{and}$$
$$\mathrm{err}(\mathcal{S}, \mu, h) = \sum_{(\mathbf{x}_i, y_i)\in\mathcal{S}} \mu(i)\chi\{h(\mathbf{x}_i) \neq y\}.$$

To learn a 1-Rule given a distribution over the training set, return the signed feature or constant with minimum weighted error. Naturally, we use the Weighted Exponential Mechanism with noise rate η to privatize selection. This is simply the Generic Private Agnostic Learner of Kasiviswanathan et al. (2011), finessing the issue that "weighted error" is actually a *set* of utility functions (analysis in Appendix C). We denote the baseline and differentially private versions of this algorithm as `1R` and `DP-1R`, respectively.

Theorem 4.1. *`DP-1R` is a $(4\eta\zeta, 0, \zeta)$-DP weak learner.*

Given a total privacy budget of ϵ, we divide it uniformly across rounds of boosting. Then, by Theorem 3.1, we solve $\epsilon = 4\tau \cdot \eta \cdot \zeta$ for η to determine how much noise `DP-1R` must inject at each round. Note that privacy depends on the statistical distance ζ between distributions over neighboring datasets. `LazyBB` furnishes the promise that $\zeta \leq 1/\kappa n$. It is natural for ζ to depend on the number of samples: the larger the dataset, the easier it is

to "hide" dependence on a single individual, and the less noise we can inject at each round. Overall:

Theorem 4.2. *LazyBB runs for τ rounds using DP-1R at noise rate $\eta = \frac{\epsilon\kappa n}{4\tau}$ is ϵ-DP.*

If a weak learning assumption holds — which for 1-Rules simplifies to "over every smooth distribution, at least one literal or constant has γ-advantage over random guessing" — then we will boost to a "good" margin. We can compute the advantage of DP-1R given this assumption.

Theorem 4.3. *Under a weak learning assumption with advantage γ, DP-1R, with probability at least $1-\beta$, is a weak learner with advantage at least $\gamma - \frac{1}{\eta}\ln\frac{|\mathcal{H}|}{\beta}$. That is, for any distribution μ over $\mathcal{S}$, we have*

$$\sum_{(\mathbf{x}_i,y_i)\in\mathcal{S}} \mu(i)\chi\{h_{out}(\mathbf{x}_i) \neq y_i\} \leq 1/2 - \left(\gamma - 1/\eta \ln |\mathcal{H}|/\beta\right)$$

where h_{out} is the output hypothesis of DP-1R.

4.2. **TopDown Decision Trees.** TopDown heuristics are a family of decision tree learning algorithms that are employed by widely used software packages such as C4.5, CART, and scikit-learn. We present a differentially private TopDown algorithm that is a modification of decision tree learning algorithms given by Kearns and Mansour (1996). At a high level, TopDown induces decision trees by repeatedly *splitting* a leaf node in the tree built so far. On each iteration, the algorithm *greedily* finds the leaf and splitting function that maximally reduces an upper bound on the error of the tree. The selected leaf is replaced by an internal node labeled with the chosen splitting function, which partitions the data at the node into two new children leaves. Once the tree is built, the leaves of the tree are labeled by the label of the most common class that reaches the leaf. Algorithm 2, DP-TopDown, is a "reference implementation" of the differentially private version of this algorithm. DP-TopDown, instead of choosing the best leaf and splitting function, applies the Exponential Mechanism to noisily select a leaf and splitting function in the built tree so far. The Exponential Mechanism is applied on the set of all possible leaves and splitting functions in the current tree; this is computationally feasible in our Boolean-feature setting. Next we introduce necessary notation and discuss the privacy guarantee of our algorithm, and how it is used as a weak learner for our boosting algorithm.

DP TopDown Decision Tree. Let F denote a class of Boolean splitting functions with input domain $\mathcal{S}$. Each internal node is labeled by a splitting function $h : \mathcal{S} \to \{0,1\}$. These splitting functions route each example $x \in \mathcal{S}$ to exactly one leaf of the tree. That is, at each internal node if the splitting function $h(x) = 0$ then x is routed to the left subtree, and x is routed to the right subree otherwise. Furthermore, let G denote the *splitting criterion*. $G : [0,1] \to [0,1]$ is a concave function which is symmetric about $1/2$ and $G(1/2) = 1$. Typical examples of splitting criterion function are Gini and Entropy. Algorithm 2 builds decision trees in which the internal nodes are labeled by functions in F, and the splitting criterion G is used to determine which leaf should be split next, and which function $h \in F$ should be used for the split.

Let T be a decision tree whose leaves are labeled by $\{0,1\}$ and μ be a distribution on $\mathcal{S}$. The weight of a leaf $\ell \in \mathit{leaves}(T)$ is defined to be the weighted fraction of data that

Algorithm 2: Differentially Private TopDown-DT

Require: Data sample $\mathcal{S}$, distribution $\hat{\mu}$ over $\mathcal{S}$, number of internal nodes t, and $\eta > 0$.

$T \leftarrow$ the single-leaf tree.
$\mathcal{C} \leftarrow \mathit{leaves}(T) \times F$
while T has fewer than t internal node **do**
  $(\ell^*, h^*) \leftarrow$ select a candidate from $\mathcal{C}$ w.p. $\propto \exp(\eta \cdot \mathrm{im}_{\ell,h,\hat{\mu}})$
  $T \leftarrow T(\ell^*, h^*)$
  for each new pair $\ell \times h \in \mathit{leaves}(T) \times F$ **do**
    $\mathrm{im}_{\ell,h,\hat{\mu}} \leftarrow G(T, \hat{\mu}) - G(T(\ell, h), \hat{\mu})$
    Add $\mathrm{im}_{\ell,h,\hat{\mu}}$ to $\mathcal{C}$
  end for
end while
Label leaves by majority label [WRNM with privacy budget $8t \cdot \eta \cdot \zeta$]
Output: T

reaches ℓ i.e., $w(\ell, \mu) = \Pr_\mu[x \text{ reaches } \ell]$. The weighted fraction of data with label 1 at leaf ℓ is denoted by $q(\ell, \mu)$. Given these we define error of T as follows.

$$\mathrm{err}(T, \mu) = \sum_{\ell \in \mathit{leaves}(T)} w(\ell, \mu) \min\{q(\ell, \mu), 1 - q(\ell, \mu)\}$$

Noting that $G(q(\ell, \mu)) \geq \min\{q(\ell, \mu), 1 - q(\ell, \mu)\}$, we have an upper bound for $\mathrm{err}(T, \mu)$.

$$\mathrm{err}(T, \mu) \leq \mathcal{G}(T, \mu) = \sum_{\ell \in \mathit{leaves}(T)} w(\ell, \mu) G(q(\ell, \mu)).$$

For $\ell \in \mathit{leaves}(T)$ and $h \in F$ let $T(\ell, h)$ denote the tree obtained from T by replacing ℓ by an internal node that splits subset of data that reaches ℓ, say $\mathcal{S}_\ell$, into two children leaves ℓ_0, ℓ_1. Note that any data x satisfying $h(x) = i$ goes to ℓ_i. The quality of a pair (ℓ, h) is the improvment we achieve by splitting at ℓ according to h. Formally,

$$\mathrm{im}_{\ell,h,\mu} = \mathcal{G}(T, \mu) - \mathcal{G}(T(\ell, h), \mu)$$

At each iteration, Algorithm 2 chooses a pair (ℓ^*, h^*) according to the Exponential Mechanism with probability proportional to $\mathrm{im}_{\ell,h,\mu}$. By Theorem 2.13, the quality of the chosen pair (ℓ^*, h^*) is close to the optimal split with high probability.

Theorem 4.4 (Privacy guarantee)**.** *`DP-TopDown`, Algorithm 2, is a $(16t \cdot \eta \cdot \zeta, 0, \zeta)$-DP weak learner.*

As before, given a total privacy budget of ϵ, we divide it uniformly across rounds of boosting. Then, by Theorem 3.1, we solve $\epsilon = 16\tau \cdot t \cdot \eta \cdot \zeta$ for η to determine how much noise `DP-TopDown` must inject at each round. `LazyBB` furnishes the promise that $\zeta \leq 1/\kappa n$. Overall:

Theorem 4.5. *`LazyBB` runs for τ rounds using `DP-TopDown` at noise rate $\eta = \frac{\epsilon \kappa n}{16\tau t}$ is ϵ-DP.*

WKL	Parameter		
	τ	λ	κ
`OneRule`	$5, 9, 15, 19, 25, 29, 39, 49, 65, 75, 99$	$0.2, 0.25, \cdots, 0.5$	$0.2, 0.25, \cdots, 0.5$
`TopDown`	$5, 9, 15, 19, 25, 29, 35, 39, 45, 51$	$0.35, 0.4$	$0.25, 0.3$

Table 1: Parameters grid

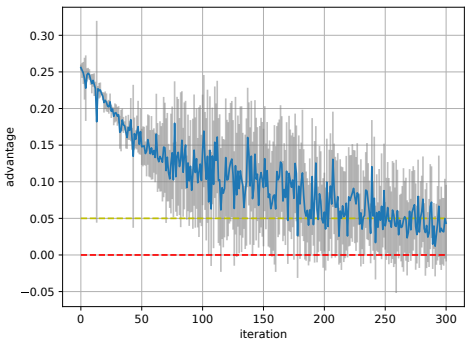

Figure 1: Advantage curve and margin histogram.

5. Experiments

Here we compare our smooth boosting algorithm (`LazyBB`) over both decision trees and 1-Rules to: differentially private logistic regression using objective perturbation (DP-LR) (Chaudhuri et al., 2011), Differentially Private Bagging (`DP-Bag`) (Jordon et al., 2019), and Privacy-Preserving Gradient Boosting Decision Trees (`DP-Boost`) (Li et al., 2020). In our implementation we used the IBM differential privacy library (available under MIT licence) (Holohan et al., 2019) for standard DP mechanisms and accounting, and scikit-learn (available under BSD licence) for infrastructure (Pedregosa et al., 2011). These experiments show that smooth boosting of *1-Rules* can yield improved model accuracy and sparsity under identical privacy constraints.

We experiment with three freely available real-world datasets. **Adult** (Available from UCI Machine Learning Repository) has 32,561 training examples, 16,282 test examples, and 162 features after dataset-oblivious one-hot coding — which incurs no privacy cost. The task is to predict if someone makes more than 50k US dollars per year from Census data. Our reported accuracies are holdout tests on the canonical test set associated with Adult. **Cod-RNA** (available from the LIBSVM website) has 59,535 training examples and 80 features after dataset-oblivious one-hot coding, and asks for detection of non-coding RNA. **Mushroom** (available from the LIBSVM website) has 8124 training examples and 117 features after one-hot coding, which asks to identify poisonous mushrooms. For Mushroom and CodRNA, we report cross-validated estimates of accuracy. All experiments were run on a 3.8 GHz 8-Core Intel Core i7 with 16GB of RAM consumer desktop computer.

Parameter Selection Without Assumptions. We select parameters for `LazyBB` and `DP-LR` entirely using grid-search and cross-validation (Table 1 for `LazyBB`) for each value of epsilon plotted i.e. $\epsilon \in (0.05, 0.1, \cdots, 0.5, 1, 3, 5)$. Note, privacy cost is not considered in our hyperparameter tuning which is also the case in the papers we compare to. Furhtemore, for different methods the hyperparameter grids are not of the same size as the hyperparameters are not comparable. However, to make sure our comparison is fair we consider large enough grid for each method, that is we set the corresponding hyperparameters too high and too low and we triggered the over-fitting and under-fitting behaviour in both cases. In particular, we tested strictly more points than Chaudhuri et al. (2011) for `DP-LR`. `DP-LR` is L_2 penalized logistic regression, with some noise added for privacy. That is, `DP-LR` caps the L_2 norm of the coefficients of a logistic regression classifier. It has a single hyperparameter C that is inversely proportional to this capped norm. Therefore, when C is large enough, `DP-LR` will recover the un-penalized, ordinary least squares solution to the logistic regression problem – up to noise added to preserve privacy. Our grid search for `DP-LR` includes C such that the coefficient norm is just as large as the ordinary least squares norm, indicating we have exhausted the meaningful range for the parameter on each dataset.

Over the small datasets we use for experiments, the Weak Learner assumption does not hold for "long enough" to realize the training error guarantee of Theorem 3.1. For example, fixing $\kappa = 1/2$ — seeking "good" margin on only half the training set — suppose we have a $(1/20)$-advantage Weak Learner. That is, at every round of boosting, each new hypothesis has accuracy at least 55% over the intermediate distribution. Under these conditions, Theorem 3.1 guarantees utility after approximately 4,000 rounds of boosting. Figure 1 plots advantage on the Adult dataset at each round of boosting with $\lambda = \gamma/4$ as required by Theorem 3.1, averaged over 10 runs of the boosting decision stumps with total privacy budget $\epsilon = 1$. The weak learner assumption fails after only 250 rounds of boosting.

And yet, even when run with much *faster* learning rate λ, we see good accuracy from `LazyBB` — the assumption holds for *small* τ, ensuring that `DP-1R` has advantage. Theorem 3.1 is much more pessimistic than is warranted. This is a know limitation of the analysis for any *non*-adaptive boosting algorithm (Schapire and Freund, 2012). In the non-private setting, we set λ very slow and boost for "many" rounds, until decay in advantage triggers a stopping criterion. In the private setting (where non-adaptivity makes differential privacy easier to guarantee) running for "many" rounds is not feasible; noise added for privacy would saturate the model. These experiments motivate further theoretical investigation of boosting dynamics for non-adaptive algorithms, due to their utility in the privacy-preserving setting.

Results. In Table 2 we plot the accuracy of each of the 5 methods above against privacy constraint ϵ, along with two non-private baselines to both quantify the "cost of privacy" and ensure that the private learners are non-trivial. The strong non-private baseline is the implementation of Gradient Boosted Trees in sklearn, the weak non-private baseline is a single 1-Rule. It is important to note that for `DP-Bag` and `DP-Boost`, we only compare our results for datasets and regimes that the corresponding hyperparameters are reported in the related works. Jordon et al. (2019) reported the performance of `DP-Bag` only on **Adult** dataset and considering low privacy regime. Li et al. (2020) reported the performance of `DP-Boost` on **Adult** and **CodRNA** datasets and only in low privacy regime. Surprisingly, we found that `LazyBB` over 1-Rules and differentially private logistic regression were the best performing models — despite being the *simplest* algorithms to state, reason about, and

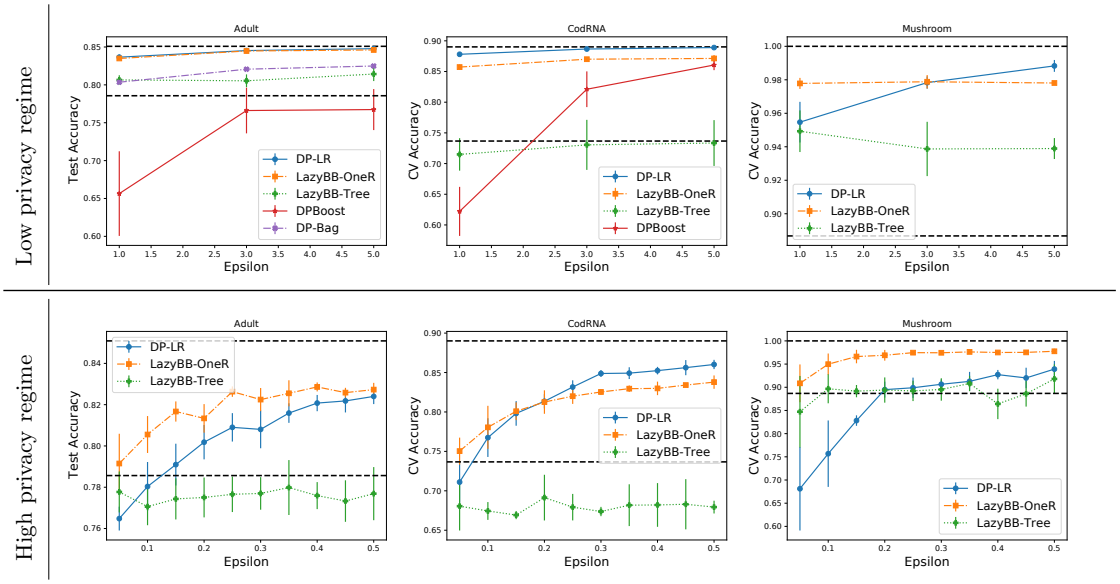

Table 2: Learning Curves — Privacy vs. accuracy.

run. (Discussions and experiments on approximate and pure differential privacy are given in Section 10.)

Sparsity, Regularization, and Interpretability. Algorithms used for high-stakes decisions should be both well-audited and privacy-preserving. However, often there is a trade-off between privacy and interpretability (Harder et al., 2020). Generally, noise injected to protect privacy harms interpretability. But our algorithms maintain accuracy under strong privacy constrains while admitting a high level of sparsity — which facilitates interpretability. Table 3 lists measurements across different levels of privacy. For an example of boosted one-rules at $\epsilon = 0.4$ DP, see Table 4.

`DP-LR` — another simple algorithm with excellent performance — uses L_2 regularization to improve generalization. While L_2 regularization keeps total mass of weights relatively small, it generally assigns non-negligible weight to *every* feature. Hence, the resulting model becomes less interpretable as the dimension of data grows. On other hand, `LazyBB` with 1-Rules controls sparsity by the number of rounds of boosting. Just as with non-private non-adaptive boosting algorithms, we can see this as a greedy approximation to L_1 regularization of a linear model Rosset et al. (2004). Moreover, the final model can be interpreted as a simple integral weighted voting of features.

We believe the use of our algorithm depends on the shape of the dataset and its multicollinearity feature rather than the datasize and privacy budget. However, our method is superior in terms of performance both in high privacy regimes and over small datasets, the question of usage boils down to the shape of the dataset at hand. It has been shown by Friedman et al. (2000) that theoretically and empirically boosting with decision stamps is a close approximation of L_1 regularization. Hence, `LazyBB` with 1-Rules is preferable in applications where sparsity is desirable and the dataset is not highly multicollinear. In fact, our experimental results give examples witnessing the common behaviour of L_1 regularization (e.g., `LazyBB` with 1-Rules) versus L_2 regularization (e.g., `DP-LR`). For instance, `DP-LR` works

ϵ	features count mean	features count std	% features
0.40	6.4	0.800	3.95%
0.50	12.8	0.400	7.90%
1.00	30.6	1.200	18.88%
3.00	72.8	2.481	44.93%
5.00	49.8	2.785	30.74%

Table 3: Statistics of number of features used by `LazyBB` with `DP-1R` across different levels of privacy on adult dataset. See the Appendix for the complete table.

votes	(feature, value)
3	marital-status : Married-civ-spouse
-2	capital-gain $= 0$
1	occupation : Exec-managerial
1	occupation : Prof-specialty
1	$13 \leq$ education-num ≤ 14.5
-1	age ≤ 17

Table 4: A 0.4-DP model obtained by training `LazyBB` with `DP-1R` on adult dataset with 0.82 accuracy.

better than `LazyBB` with 1-Rules in **Cod-RNA**, but not in **Mushroom**. For the complete table of sparsity measurements for all the experiments see Appendix A.

Pessimistic Generalization Theory. Empirically, `LazyBB` generalizes well. As with AdaBoost, we could try to explain this with large margins and Rademacher complexity, which applies to any voting classifier. We estimated the Rademacher complexity of 1-Rules over each dataset to predict test error. The bounds are far more pessimistic than the experiments; please see the Appendix for comparison tables. Intuitively, if `LazyBB` showed larger margins on the training data than on unseen data, this would constitute a *membership inference attack* — which is ruled out by differential privacy. This motivates theoretical investigation of new techniques to guarantee generalization of differentially private models trained on small samples.

6. Weighted Exponential Mechanism: Proof of Theorem 2.13

In this section we discuss the privacy guarantee of the Weighted Exponential Mechanism defined in Definition 2.12. Our proof follows the same steps as the standard Exponential Mechanism (McSherry and Talwar, 2007). Our goal is to prove, given two neighboring datasets and two similar distributions on them, the Weighted Exponential Mechanism outputs the same hypothesis with high probability.

In what follows let M denote the Weighted Exponential Mechanism, and let $\mathcal{H} = \text{Range}(M)$. Suppose $\mathcal{S}, \mathcal{S}'$ are two neighboring datasets of size n and μ, μ' are distributions over $[n]$ such that $\mathtt{d}(\mu, \mu') < \zeta$. Furthermore, let $q_{D,\mu} \colon \mathcal{H} \to \mathbb{R}$ be a quality score that has

robust sensitivity Δ_ζ. That is, for every hypothesis $h \in \mathcal{H}$, we have

$$\max_{\substack{D \sim D' \\ \mu,\mu' : \mathtt{d}(\mu,\mu') < \zeta}} |q_{D,\mu}(h) - q_{D',\mu'}(h)| \leq \Delta_\zeta. \tag{6.1}$$

We proceed to prove that for any $h \in \mathcal{H}$ the following holds

$$\Pr[M(S,\mu) = h] \leq \exp(2\eta\Delta_\zeta) \Pr[M(S',\mu') = h].$$

Recall that M outputs a hypothesis h with probability proportional to $\exp(\eta \cdot q_{D,\mu})$ with $\eta = \frac{\epsilon}{2\Delta_\zeta}$. Let us expand the probabilities above,

$$\frac{\Pr[M(\mathcal{S},\mu) = h]}{\Pr[M(\mathcal{S}',\mu') = h]} = \frac{\exp\left(\eta \cdot q_{\mathcal{S},\mu}(h)\right)}{\exp\left(\eta \cdot q_{\mathcal{S}',\mu'}(h)\right)} \times \frac{\sum\limits_{h \in \mathcal{H}} \exp\left(\eta \cdot q_{\mathcal{S}',\mu'}(h)\right)}{\sum\limits_{h \in \mathcal{H}} \exp\left(\eta \cdot q_{\mathcal{S},\mu}(h)\right)}.$$

Consider the first term, then

$$\frac{\exp\left(\eta \cdot q_{\mathcal{S},\mu}(h)\right)}{\exp\left(\eta \cdot q_{\mathcal{S}',\mu'}(h)\right)} = \exp\left(\eta[q_{\mathcal{S},\mu}(h) - q_{\mathcal{S}',\mu'}(h)]\right) \leq \exp\left(\eta \cdot \Delta_\zeta\right) \qquad \text{(By (6.1))}$$

Now consider the second term, then

$$\begin{aligned}
\frac{\sum\limits_{h \in \mathcal{H}} \exp\left(\eta \cdot q_{\mathcal{S}',\mu'}(h)\right)}{\sum\limits_{h \in \mathcal{H}} \exp\left(\eta \cdot q_{\mathcal{S},\mu}(h)\right)} &\leq \frac{\sum\limits_{h \in \mathcal{H}} \exp\left(\eta \cdot [q_{\mathcal{S},\mu}(h) + \Delta_\zeta]\right)}{\sum\limits_{h \in \mathcal{H}} \exp\left(\eta \cdot q_{\mathcal{S},\mu}(h)\right)} \\
&= \frac{\exp\left(\eta\Delta_\zeta\right) \sum\limits_{h \in \mathcal{H}} \exp\left(\eta \cdot q_{\mathcal{S},\mu}(h)\right)}{\sum\limits_{h \in \mathcal{H}} \exp\left(\eta \cdot q_{\mathcal{S},\mu}(h)\right)} \\
&= \exp\left(\eta\Delta_\zeta\right)
\end{aligned}$$

Hence, it follows that

$$\frac{\Pr[M(\mathcal{S},\mu) = h]}{\Pr[M(\mathcal{S}',\mu') = h]} \leq \exp\left(\eta\Delta_\zeta\right) \cdot \exp\left(\eta\Delta_\zeta\right) = \exp\left(2\eta\Delta_\zeta\right)$$

This implies that, for $\eta > 0$, WEM is a $(2\eta\Delta_\zeta, 0, \zeta)$-differentially private weak learner. (Note that setting $\eta = \frac{2\epsilon}{2\Delta_\zeta}$ yields a $(\epsilon, 0, \zeta)$-differentially private weak learner.)

We point out that the proof for the utility guarantee of Theorem 2.13 is identical to the proof of the utility guarantee in standard Exponential Mechanism (McSherry and Talwar, 2007).

7. Weighted Return Noisy Max: Proof of Theorem 2.14

In this section we discuss the privacy guarantee of the Weighted Return Noisy Max defined in Section 2.4. Our proof follows the same steps as the standard Return Noisy Max explained in Dwork and Roth (2014) with slight modification. Our goal is to prove, given two neighboring datasets and two similar distributions on them, the WRNM outputs the same hypothesis index.

Let $f_1, \ldots, f_k$ be k quality functions where each $f_i : \mathcal{S} \times \mathcal{D}(S) \to \mathbb{R}$ maps datasets and distributions over them to real numbers. For a dataset S and distribution μ over S, WRNM adds independently generated Laplace noise $Lap(1/\eta)$ to each f_i and returns the

index of the largest noisy function i.e. $i^* = \operatorname{argmax}_i(f_i + Z_i)$ where each Z_i denotes a random variable drawn independently from the Laplace distribution with scale parameter $1/\eta$. In what follows let M denote the WRNM.

Suppose $\mathcal{S}, \mathcal{S}'$ are two neighboring datasets of size n and μ, μ' are distributions over $[n]$ such that $\mathtt{d}(\mu, \mu') < \zeta$. Furthermore, suppose each f_i has robust sensitivity at most Δ_ζ. That is, for every index $i \in \{1, \ldots, k\}$, we have

$$\max_{\substack{D \sim D' \\ \mu, \mu' : \mathtt{d}(\mu,\mu') < \zeta}} |f_i(\mathcal{S}, \mu) - f_i(\mathcal{S}', \mu')| \leq \Delta_\zeta. \tag{7.1}$$

Fix any $i \in \{1, \ldots, k\}$. We will bound the ratio of the probabilities that i is selected by M with inputs $\mathcal{S}, \mathcal{S}'$ and distributions μ, μ'.

Fix $Z_{-i} = (Z_1, \ldots, Z_{i-1}, Z_{i+1}, \ldots, Z_k)$, where each $Z_j \in Z_{-i}$ is drawn from $Lap(1/\eta)$. We first argue that

$$\frac{\Pr[M(\mathcal{S}, \mu) = i \mid Z_{-i}]}{\Pr[M(\mathcal{S}', \mu') = i \mid Z_{-i}]} \leq \mathrm{e}^{2\eta \cdot \Delta_\zeta}.$$

Define Z^* to be the minimum Z_i such that

$$f_i(\mathcal{S}, \mu) + Z^* > f_j(\mathcal{S}, \mu) + Z_j \ \forall j \neq i$$

Note that, having fixed Z_{-i}, M will output i only if $Z_i \geq Z^*$. Recalling (7.1), for all $j \neq i$, we have the following,

$$f_i(\mathcal{S}', \mu') + Z^* + \Delta_\zeta \geq f_i(\mathcal{S}, \mu) + Z^* > f_j(\mathcal{S}, \mu) + Z_j \geq f_j(\mathcal{S}', \mu') + Z_j - \Delta_\zeta$$

This implies that

$$f_i(\mathcal{S}', \mu') + Z^* + 2\Delta_\zeta \geq f_j(\mathcal{S}', \mu') + Z_j$$

Now, for dataset $\mathcal{S}'$, distribution μ', and Z_{-i}, mechanism M selects the i-th index if Z_i, drawn from $Lap(1/\eta)$, satisfies $Z_i \geq Z^* + 2\Delta_\zeta$.

$$\begin{aligned}
\Pr_{Z_i \sim Lap(1/\eta)}[M(\mathcal{S}', \mu') = i \mid Z_{-i}] &\geq \Pr_{Z_i \sim Lap(1/\eta)}[Z_i \geq Z^* + 2\Delta_\zeta] \\
&\geq \mathrm{e}^{-(2\eta\Delta_\zeta)} \Pr_{Z_i \sim Lap(1/\eta)}[Z_i \geq Z^*] \\
&= \mathrm{e}^{-(2\eta\Delta_\zeta)} \Pr_{Z_i \sim Lap(1/\eta)}[M(\mathcal{S}, \mu) = i \mid Z_{-i}]
\end{aligned}$$

Multiplying both sides by $\mathrm{e}^{(2\eta\Delta_\zeta)}$ yields the desired bound.

$$\frac{\Pr_{Z_i \sim Lap(1/\eta)}[M(\mathcal{S}, \mu) = i \mid Z_{-i}]}{\Pr_{Z_i \sim Lap(1/\eta)}[M(\mathcal{S}', \mu') = i \mid Z_{-i}]} \leq \mathrm{e}^{(2\eta\Delta_\zeta)}$$

This implies that, for $\eta > 0$, WRNM is a $(2\eta\Delta_\zeta, 0, \zeta)$-differentially private weak learner. (Note that setting $\eta = \frac{2\epsilon}{2\Delta_\zeta}$ yields a $(\epsilon, 0, \zeta)$-differentially private weak learner.)

Algorithm 3: Differentially Private 1-Rule Induction($\mathcal{S}, \mu, \eta$)

Require: Dataset $\mathcal{S}$, distribution μ over $[1, \ldots, |\mathcal{S}|]$, and $\eta > 0$.

1: Let $\mathcal{H}$ be the set of all literals over $\mathcal{S}$ plus the constants True and False
2: **for** $h \in \mathcal{H}$ **do**
3: $\quad q_{\mathcal{S},\mu}(h) \leftarrow -\mathrm{err}(\mathcal{S}, \mu, h)$.
4: **end for**
5: $h_{out} \leftarrow$ select a hypothesis $h \in \mathcal{H}$ with probability proportional to $\exp(\eta \cdot q_{\mathcal{S},\mu}(h))$
6: **return** h_{out}

8. Weak Learner: DP 1-Rules

Throughout this section let $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}^n$ where $\mathbf{x}_i = (x_{i1}, \ldots, x_{ir})$ denote a dataset, and let μ be a distribution over $[n]$. We will brute-force "1-Rules," also known as Decision Stumps (Iba and Langley, 1992; Holte, 1993). Here, these simply evaluate a single Boolean literal such as $\neg x_{17}$ — an input variable that may or may not be negated. We also admit the constants True and False as literals.

A brutally simple but surprisingly effective weak learner returns the literal with optimal weighted agreement to the labels. For any 1-Rule h define $\mathrm{err}(\mathcal{S}, \mu, h)$ to be:

$$\mathrm{err}(\mathcal{S}, \mu, h) = \sum_{(\mathbf{x}_i, y_i) \in \mathcal{S}} \mu(i) \chi\{h(\mathbf{x}_i) \neq y\}.$$

For learning 1-Rules under DP constraints, the natural approach is to use the Exponential Mechanism to noisily select the best possible literal. There is a small type error: the standard Exponential Mechanism does not consider utility functions with an auxiliary weighting μ. But for weak learning we only demand privacy (close output distributions) when *both* the dataset and measures are "close." When both promises hold and μ is fixed, the Exponential Mechanism is indeed a differentially private 1-Rule learner. We show this formally below.

Observation 8.1. Let $\mathcal{S} \sim \mathcal{S}'$ be any two neighboring datasets and set $I = \mathcal{S} \cap \mathcal{S}'$. Then, for any two distributions μ, μ' over $[n]$, we have

$$\begin{aligned}
&\left|\mathrm{err}(\mathcal{S}, \mu, T) - \mathrm{err}(\mathcal{S}', \mu', T)\right| \\
&= \left| \sum_{(\mathbf{x}_i, y_i) \in \mathcal{S}} \mu(i) \chi\{T(\mathbf{x}_i) \neq y\} - \sum_{(\mathbf{x}_i, y_i) \in \mathcal{S}'} \mu'(i) \chi\{T(\mathbf{x}_i) \neq y\} \right| \\
&= \left| \sum_{(\mathbf{x}_i, y_i) \in \mathcal{S} \cap \mathcal{S}'} [\mu(i) - \mu'(i)] \chi\{T(\mathbf{x}_i) \neq y\} + \sum_{(\mathbf{x}_i, y_i) \in \mathcal{S} \triangle \mathcal{S}'} [\mu(i) - \mu'(i)] \chi\{T(\mathbf{x}_i) \neq y\} \right| \\
&\leq \sum_{(\mathbf{x}_i, y_i) \in \mathcal{S} \cap \mathcal{S}'} |\mu(i) - \mu'(i)| + \sum_{(\mathbf{x}_i, y_i) \in \mathcal{S} \triangle \mathcal{S}'} |\mu(i) - \mu'(i)| \\
&= \sum_{i=1}^{n} |\mu(i) - \mu'(i)| = 2\mathtt{d}(\mu, \mu').
\end{aligned}$$

Theorem 8.2. *Algorithm 3 is a $(4\eta\zeta, 0, \zeta)$-differentially private weak learner.*

Proof. Suppose $\mathcal{S}, \mathcal{S}'$ are two neighboring datasets of size n and μ, μ' are distributions over $[n]$ such that $\mathtt{d}(\mu, \mu') < \zeta$. Observation 8.1 tells us that the quality score $q_{\mathcal{S},\mu}(h) = \mathrm{err}(\mathcal{S}, \mu, h)$

has robust sensitivity 2ζ. Hence, by Theorem 2.13, we have that Algorithm 3 is a $(4\eta\zeta, 0, \zeta)$-differentially private weak learner. □

Theorem 8.3. *Let h_{opt} denote the optimal hypothesis in $\mathcal{H}$. Then Algorithm 3, with probability at least $1-\beta$, returns $h_{out} \in \mathcal{H}$ such that*

$$\operatorname{err}(h_{out}) \leq \operatorname{err}(h_{opt}) + \frac{1}{\eta} \ln \frac{|\mathcal{H}|}{\beta}.$$

Proof. By Theorem 2.13, with probability at least $1-\beta$, we have

$$q_{\mathcal{S},\mu}(h_{out}) \geq \max_{h \in \mathcal{H}} q_{\mathcal{S},\mu}(h) - \frac{1}{\eta} \ln \frac{|\mathcal{H}|}{\beta} \tag{8.1}$$

Note that $q_{\mathcal{S},\mu}(h) = -\operatorname{err}(\mathcal{S}, \mu, h)$ for all $h \in \mathcal{H}$ and $\max_{h \in \mathcal{H}} q_{\mathcal{S},\mu}(h) = -\operatorname{err}(h_{opt})$. This gives us

$$\begin{aligned} -\operatorname{err}(h_{out}) &\geq -\operatorname{err}(h_{opt}) - \frac{1}{\eta} \ln \frac{|H|}{\beta} \implies \\ \operatorname{err}(h_{out}) &\leq \operatorname{err}(h_{opt}) + \frac{1}{\eta} \ln \frac{|\mathcal{H}|}{\beta}. \end{aligned}$$

□

As we already discussed, in order to construct PAC learners by boosting weak learners we need weak learners that only beat random guessing on any distribution over the training set. Here, we wish to use Algorithm 3 as a weak learner. That is, we show that Algorithm 3 (with high probability) is better than random guessing. In what follows we have Theorem 4.3 and its proof.

Theorem 8.4. *Under a weak learner assumption with advantage γ, Algorithm 3, with probability at least $1-\beta$, is a weak learner with advantage at least $\gamma - \frac{1}{\eta} \ln \frac{|\mathcal{H}|}{\beta}$. That is, for any distribution μ over $\{1, \ldots, |\mathcal{S}|\}$, we have*

$$\sum_{(\mathbf{x}_i, y_i) \in \mathcal{S}} \mu(i) \chi\{h_{out}(\mathbf{x}_i) \neq y\} \leq 1/2 - \left(\gamma - \frac{1}{\eta} \ln \frac{|\mathcal{H}|}{\beta}\right).$$

Proof. By Theorem 8.3, Algorithm 3 with probability at least $1-\beta$ outputs a hypothesis h_{out} such that

$$\operatorname{err}(h_{out}) \leq \operatorname{err}(h_{opt}) + \frac{1}{\eta} \ln \frac{|\mathcal{H}|}{\beta}.$$

Under a *weak learner assumption*, we assume that an optimal hypothesis h_{opt} is at least as good as random guessing. That is $\operatorname{err}(h_{opt}) < 1/2 - \gamma$. This yields the desired result. □

9. Proof of Theorem 4.4

Here we consider splitting criterion to be the Gini criterion $G(q) = 4q(1-q)$. Note that this function is symmetric about $1/2$ and $G(1/2) = 1$. Throughout this section, $\mathcal{S} \sim \mathcal{S}'$ are two neighboring datasets of size n and μ, μ' are distributions over $[n]$ such that $\mathtt{d}(\mu, \mu') < \zeta$. Observe that for a decision tree T we have $|w(\ell, \mu) - w(\ell, \mu')| \leq \zeta$ and $|q(\ell, \mu) - q(\ell, \mu')| \leq \zeta$. Before proceeding to provide an upper bound on the sensitivity of $\mathrm{im}_{\ell,h,\mu}$, we prove some useful lemmas.

Lemma 9.1. *The following holds.* $4\Big|w(\ell,\mu)q(\ell,\mu)(1-q(\ell,\mu)) - w(\ell,\mu')q(\ell,\mu)(1-q(\ell,\mu'))\Big| \leq \frac{5}{4}\zeta$

Proof. As the Gini criterion $G(q) = 4q(1-q)$ is symmetric about $1/2$, without loss of generality, we assume $q(\ell) \leq 1/2$. Furthermore, suppose $w(\ell,\mu)q(\ell,\mu)(1-q(\ell,\mu))$ is greater than $w(\ell,\mu')q(\ell,\mu')(1-q(\ell,\mu'))$. The arguments for the other cases are analogous.

$$
\begin{aligned}
&w(\ell,\mu)q(\ell,\mu)(1-q(\ell,\mu)) - w(\ell,\mu')q(\ell,\mu')(1-q(\ell,\mu')) \\
&\leq w(\ell,\mu)q(\ell,\mu)(1-q(\ell,\mu)) - w(\ell,\mu')(q(\ell,\mu)-\zeta)(1-q(\ell,\mu)+\zeta) \\
&= w(\ell,\mu)q(\ell,\mu)(1-q(\ell,\mu)) - w(\ell,\mu')q(\ell,\mu)(1-q(\ell,\mu)+\zeta) + w(\ell,\mu')\zeta(1-q(\ell,\mu)+\zeta) \\
&\leq w(\ell,\mu)q(\ell,\mu)(1-q(\ell,\mu)) - w(\ell,\mu')q(\ell,\mu)(1-q(\ell,\mu)) + \zeta \\
&\leq |w(\ell,\mu) - w(\ell,\mu')|q(\ell,\mu)(1-q(\ell,\mu)) + \zeta \leq \frac{5}{4}\zeta
\end{aligned}
$$

□

Lemma 9.2. *For a decision tree T and $(\ell,h) \in leaves(T) \times F$ we have*

$$\Big|\mathrm{im}_{\ell,h,\mu}(\mathcal{S}) - \mathrm{im}_{\ell,h,\mu'}(\mathcal{S}')\Big| \leq 4\zeta.$$

Proof. For dataset $\mathcal{S}$ let $\mathcal{G}(T) = \sum_{\ell \in leaves(T)} w(\ell)G(q(\ell))$. Recall the definition of $\mathrm{im}_{\ell,\mathrm{h},\mu}$,

$$
\begin{aligned}
\mathrm{im}_{\ell,h,\mu}(\mathcal{S}) &= \mathcal{G}(T,\mu) - \mathcal{G}(T(\ell,h),\mu) \\
&= w(\ell,\mu)G(q(\ell,\mu)) - w(\ell_0,\mu)G(q(\ell_0,\mu)) - w(\ell_1,\mu)G(q(\ell_1,\mu))
\end{aligned}
$$

Similarly, for dataset $\mathcal{S}'$ let $\mathcal{G}(T,\mu') = \sum_{\ell \in leaves(T)} w(\ell,\mu')G(q(\ell,\mu'))$. Then we have

$$
\begin{aligned}
&\mathrm{im}_{\ell,h,\mu'}(\mathcal{S}') = \mathcal{G}(T,\mu') - \mathcal{G}(T(\ell,h),\mu') \\
&= w(\ell,\mu')G(q(\ell,\mu')) - w(\ell_0,\mu')G(q(\ell_0,\mu')) - w(\ell_1,\mu')G(q(\ell_1,\mu'))
\end{aligned}
$$

Having these we can rewrite $\Big|\mathrm{im}_{\ell,h,\mu}(\mathcal{S}) - \mathrm{im}_{\ell,h,\mu'}(\mathcal{S}')\Big|$ as follows,

$$
\begin{aligned}
&\Big|\mathrm{im}_{\ell,h,\mu}(\mathcal{S}) - \mathrm{im}_{\ell,h,\mu'}(\mathcal{S}')\Big| \\
&= \Big|\mathcal{G}(T,\mu) - \mathcal{G}(T(\ell,h),\mu) - \mathcal{G}(T,\mu') + \mathcal{G}(T(\ell,h),\mu')\Big| \\
&= \Big|w(\ell,\mu)G(q(\ell,\mu)) - w(\ell_0,\mu)G(q(\ell_0,\mu)) - w(\ell_1,\mu)G(q(\ell_1,\mu)) - w(\ell,\mu')G(q(\ell,\mu')) \\
&\quad + w(\ell_0,\mu')G(q(\ell_0,\mu')) + w(\ell_1,\mu')G(q(\ell_1,\mu'))\Big| \\
&\leq \Big|w(\ell,\mu)G(q(\ell,\mu)) - w(\ell,\mu')G(q(\ell,\mu'))\Big| + \Big|w(\ell_0,\mu')G(q(\ell_0,\mu')) - w(\ell_0,\mu)G(q(\ell_0,\mu))\Big| \\
&\quad + \Big|w(\ell_1,\mu')G(q(\ell_1,\mu')) - w(\ell_1,\mu)G(q(\ell_1,\mu))\Big| \\
&\leq 15/4\zeta \\
&\leq 4\zeta
\end{aligned}
$$

where the last inequalities follow by Lemma 9.1. □

Let us denote Algorithm 2 by M. Consider a fix decision tree T. We prove that, given $\mathcal{S} \sim \mathcal{S}'$ and μ, μ', Algorithm 2 chooses the same leaf and split function with high probability.

Let $\mathcal{C} = leaves(T) \times F$ denote the set of possible split candidates. For each $(\ell, h) \in \mathcal{C}$, $\mathrm{im}_{\ell,h,\mu}(\mathcal{S})$ denotes the improvement gained in classification of dataset $\mathcal{S}$ by splitting T at leaf ℓ according to split function h. Similarly, we have $\mathrm{im}_{\ell,h,\mu'}(\mathcal{S}')$. Provided that $\mathtt{d}(\mu, \mu') \leq \zeta$, by Lemma 9.2, the robust sensitivity of quality score $\mathrm{im}_{\ell,h,\mu}$ is at most 4ζ. Similar to the proof of Theorem 2.13 it follows that

$$\frac{\Pr[M(\mathcal{S}, \mu) = (\ell, h)]}{\Pr[M(\mathcal{S}', \mu') = (\ell, h)]} \leq \exp\left(8 \cdot \eta \cdot \zeta\right).$$

This means each selection procedure where `DP-TopDown` selects a leaf and a splitting function is $(8 \cdot \eta \cdot \zeta, 0, \zeta)$-differentially private. Using the composition theorems for differentially private mechanisms, Theorem 2.8, yields privacy guarantee

$$\tilde{\epsilon} = 8t \cdot \eta \cdot \zeta$$

for the construction of the internal nodes. We use $\tilde{\epsilon}$ for labeling the leaves using Laplace Mechanism. Since the leaves partition dataset, this preserves $\tilde{\epsilon}$-differential privacy by parallel composition of deferentially private mechanisms (Theorem 2.9). Overall, `TopDown-DT` is an $(16t \cdot \eta \cdot \zeta, 0, \zeta)$-differentially private weak learner.

Remark 9.3. Using advanced composition for differentially private mechanisms, Theorem 2.8, for every $\tilde{\delta} > 0$ yields privacy guarantee

$$\tilde{\epsilon}_{\tilde{\delta}} = t(8 \cdot \eta \cdot \zeta)^2 + 8 \cdot \eta \cdot \zeta \sqrt{t \log(1/\tilde{\delta})}$$

for the construction of the internal nodes. We use $\tilde{\epsilon}_{\tilde{\delta}}$ for labeling the leaves using Laplace Mechanism. Since the leaves partition dataset, this preserves $\tilde{\epsilon}_{\tilde{\delta}}$-differential privacy by parallel composition of deferentially private mechanisms (Theorem 2.9). Overall, `TopDown-DT` is an $(2\tilde{\epsilon}_{\tilde{\delta}}, \tilde{\delta}, \zeta)$-differentially private weak learner.

10. Approximate Differential Privacy

Figures 2 and 3 compare the cross validation average accuracy on **Adult** dataset in the pure and approximate differential privacy regimes, for two different strategies of hyperparemeter selection: oblivious to ϵ (Figure 2), and ϵ-dependent (Figure 3). This emphasizes the importance of tuning hyperparameters for each choice of ϵ *separately*. For approximate differential privacy, we consider the small constant value of $\delta = 10^{-5}$, the same as that used by `DP-Bag` (Jordon et al., 2019).

When we set hyperparameters identically for each ϵ, using approximate differential privacy can allow significantly increased accuracy at each ϵ. We found this to be the case especially for higher τ; we select $\tau = 99$ to illustrate. However, if we are allowed to separately optimize for each ϵ, the significance of this advantage disappears. Though average accuracy clearly improves, it is not outside one standard deviation of average accuracy for pure differential privacy. It seems that boosted 1-Rules are too simple to distinguish between pure and approximate differential privacy constraints on this small dataset.

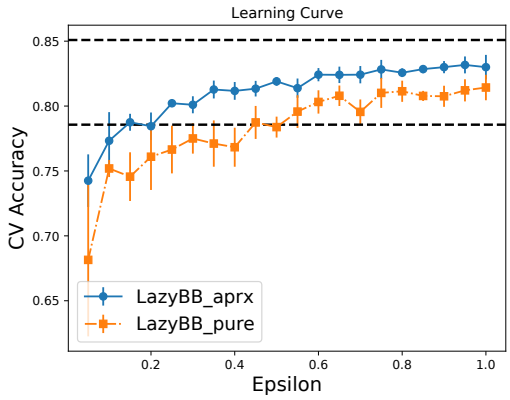

Figure 2: CV accuracy on Adult of (ϵ, δ)-DP `LazyBB` ($\kappa = 1/4,\ \lambda = 1/4,\ \tau = 99$) with `DP-1R`, $\delta \in \{0, 10^{-5}\}$, varying ϵ, vs. non-private baselines.

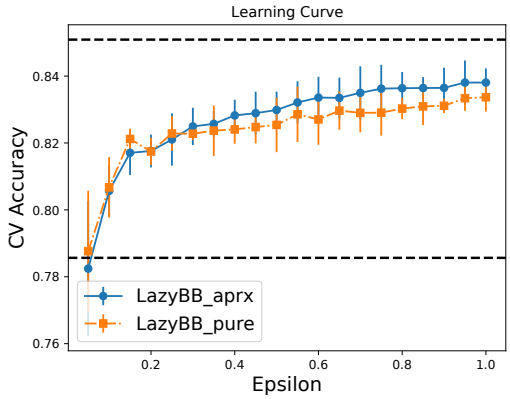

Figure 3: CV accuracy on Adult of (ϵ, δ)-DP `LazyBB` with `DP-1R`, $\delta \in \{0, 10^{-5}\}$, varying ϵ, vs. non-private baselines, with best model for each ϵ displayed.

References

B. Barak, M. Hardt, and S. Kale. The uniform hardcore lemma via approximate bregman projections. In C. Mathieu, editor, *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1193–1200. SIAM, 2009. http://dl.acm.org/citation.cfm?id=1496770.1496899.

A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: the SuLQ framework. In C. Li, editor, *Proceedings of the 24th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 128–138. ACM, 2005. https://doi.org/10.1145/1065167.1065184.

L. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200–217, 1967. ISSN 0041-5553. https://doi.org/https://doi.org/10.1016/0041-5553(67)90040-7.

L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984. ISBN 0-534-98053-8.

M. Bun and T. Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In M. Hirt and A. D. Smith, editors, *Proceedings of Theory of Cryptography Conference (TCC-B)*, volume 9985 of *Lecture Notes in Computer Science*, pages 635–658, 2016. https://doi.org/10.1007/978-3-662-53641-4_24.

M. Bun, M. L. Carmosino, and J. Sorrell. Efficient, noise-tolerant, and private learning via boosting. In J. D. Abernethy and S. Agarwal, editors, *Proceedings of Conference on Learning Theory (COLT)*, volume 125 of *Proceedings of Machine Learning Research*, pages 1031–1077. PMLR, 2020. http://proceedings.mlr.press/v125/bun20a.html.

K. Chaudhuri, C. Monteleoni, and A. D. Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12:1069–1109, 2011. http://dl.acm.org/citation.cfm?id=2021036.

C. Dwork. Differential privacy. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors, *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP)*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2006. https://doi.org/10.1007/11787006_1.

C. Dwork and J. Lei. Differential privacy and robust statistics. In M. Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC)*, pages 371–380. ACM, 2009. https://doi.org/10.1145/1536414.1536466.

C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In S. Vaudenay, editor, *Proceedings of the 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, volume 4004 of *Lecture Notes in Computer Science*, pages 486–503. Springer, 2006. https://doi.org/10.1007/11761679_29.

C. Dwork, G. N. Rothblum, and S. P. Vadhan. Boosting and differential privacy. In *Proceedings of the 51th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 51–60. IEEE Computer Society, 2010. https://doi.org/10.1109/FOCS.2010.12.

A. Friedman and A. Schuster. Data mining with differential privacy. In B. Rao, B. Krishnapuram, A. Tomkins, and Q. Yang, editors, *Proceedings of the 16th ACM SIGKDD*

International Conference on Knowledge Discovery and Data Mining, pages 493–502. ACM, 2010. https://doi.org/10.1145/1835804.1835868.

J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2): 337–407, 2000.

F. Harder, M. Bauer, and M. Park. Interpretable and differentially private predictions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:4083–4090, Apr. 2020. https://doi.org/10.1609/aaai.v34i04.5827.

N. Holohan, S. Braghin, P. M. Aonghusa, and K. Levacher. Diffprivlib: The IBM differential privacy library. *CoRR*, abs/1907.02444, 2019. http://arxiv.org/abs/1907.02444.

R. C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11:63–91, 1993. https://doi.org/10.1023/A:1022631118932.

L. Hyafil and R. L. Rivest. Constructing optimal binary decision trees is NP-Complete. *Information Processing Letters*, 5(1):15–17, 1976. https://doi.org/10.1016/0020-0190(76)90095-8.

W. Iba and P. Langley. Induction of one-level decision trees. In D. H. Sleeman and P. Edwards, editors, *Proceedings of the Ninth International Workshop on Machine Learning*, pages 233–240. Morgan Kaufmann, 1992. https://doi.org/10.1016/b978-1-55860-247-2.50035-8.

J. Jordon, J. Yoon, and M. van der Schaar. Differentially private bagging: Improved utility and cheaper privacy than subsample-and-aggregate. In *Advances in Neural Information Processing Systems 32*, pages 4325–4334, 2019. https://proceedings.neurips.cc/paper/2019/hash/5dec707028b05bcbd3a1db5640f842c5-Abstract.html.

S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. D. Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011. https://doi.org/10.1137/090756090.

M. J. Kearns and Y. Mansour. On the boosting ability of top-down decision tree learning algorithms. In G. L. Miller, editor, *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing*, pages 459–468. ACM, 1996. https://doi.org/10.1145/237814.237994.

D. A. Levin and Y. Peres. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017.

Q. Li, Z. Wu, Z. Wen, and B. He. Privacy-preserving gradient boosting decision trees. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 784–791. AAAI Press, 2020. https://aaai.org/ojs/index.php/AAAI/article/view/5422.

F. McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. *Commununications of the ACM*, 53(9):89–97, 2010. https://doi.org/10.1145/1810891.1810916.

F. McSherry and K. Talwar. Mechanism design via differential privacy. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 94–103. IEEE Computer Society, 2007. https://doi.org/10.1109/FOCS.2007.41.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986. https://doi.org/10.1023/A:1022643204877.

J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993. ISBN 1-55860-238-0.

L. Rokach and O. Maimon. *Data Mining with Decision Trees*. WORLD SCIENTIFIC, 2nd edition, 2014. https://doi.org/10.1142/9097.

S. Rosset, J. Zhu, and T. Hastie. Boosting as a regularized path to a maximum margin classifier. *Journal of Machine Learning Research*, 5:941–973, 2004. http://jmlr.org/papers/volume5/rosset04a/rosset04a.pdf.

R. E. Schapire and Y. Freund. *Boosting: Foundations and Algorithms*. The MIT Press, 2012. ISBN 0262017180.

K. Wang, T. Dick, and M. Balcan. Scalable and provably accurate algorithms for differentially private distributed decision tree learning. *CoRR*, abs/2012.10602, 2020. https://arxiv.org/abs/2012.10602.

A. Wood, M. Altman, A. Bembenek, M. Bun, M. Gaboardi, J. Honaker, K. Nissim, D. R. OBrien, T. Steinke, and S. Vadhan. Differential privacy: A primer for a non-technical audience. *Vanderbilt Journal of Entertainment & Technology Law*, 21(1):209–275, 2018.

Appendix A. Sparsity statistics of the experiments

In Section 5, we discussed sparsity and interpretability of `LazyBB` with 1-Rules. Here we share the complete table of sparsity measurements for all the experiments. For each level of privacy, we use the hyper-parameter selected by cross-validation and repeated the experiment 5 times to obtain confidence bounds.

ϵ	features count mean	features count std	% features
0.05	4.6	0.489	2.83%
0.10	4.8	0.400	2.96%
0.15	3.8	0.400	2.34%
0.20	3.6	1.019	2.22%
0.25	7.0	0.632	4.32%
0.30	13.8	1.166	8.51%
0.35	7.6	0.489	4.69%
0.40	6.4	0.800	3.95%
0.45	19.2	2.785	11.85%
0.50	12.8	0.400	7.90%
1.00	30.6	1.200	18.88%
3.00	72.8	2.481	44.93%
5.00	49.8	2.785	30.74%

Table 5: Sparsity measurements for Adult dataset.

ϵ	features count mean	features count std	% features
0.05	6.0	0.632	7.50%
0.10	12.4	1.744	15.50%
0.15	19.6	1.625	24.50%
0.20	16.8	1.327	21.00%
0.25	11.2	0.748	14.00%
0.30	10.2	0.748	12.75%
0.35	26.4	1.356	33.00%
0.40	19.8	2.482	24.75%
0.45	34.4	1.497	43.00%
0.50	25.4	2.653	31.75%
1.00	54.2	3.187	67.75%
3.00	44.2	2.227	55.25%
5.00	32.0	2.098	40.00%

Table 6: Sparsity measurements for Cod-RNA dataset.

ϵ	features count mean	features count std	% features
0.05	4.6	0.490	3.93%
0.10	7.2	1.166	6.15%
0.15	5.8	0.748	4.95%
0.20	8.6	1.497	7.35%
0.25	6.2	0.748	5.29%
0.30	5.6	0.490	4.78%
0.35	9.0	0.894	7.69%
0.40	9.8	1.166	8.37%
0.45	9.4	1.356	8.03%
0.50	11.8	1.720	10.08%
1.00	14.4	1.625	12.03%
3.00	28.8	2.926	24.61%
5.00	11.8	0.748	10.08%

Table 7: Sparsity measurements for Mushroom dataset.

Appendix B. Hyperparameters

These are the hyperparemeters selected by cross-validation of boosted 1-Rules over each of our datasets. The privacy vs. accuracy curves use these settings for each value of ϵ.

ϵ	density	learning rate	no. estimators
0.05	0.50	0.50	5
0.10	0.45	0.50	5
0.15	0.50	0.40	5
0.20	0.50	0.30	5
0.25	0.35	0.50	9
0.30	0.40	0.40	19
0.35	0.30	0.45	9
0.40	0.35	0.50	9
0.45	0.40	0.45	25
0.50	0.35	0.50	15
1.00	0.35	0.45	39
3.00	0.35	0.45	99
5.00	0.35	0.45	75

Table 8: Hyperparameters selected by cross-validation for Adult dataset.

ϵ	density	learning rate	no. estimators
0.05	0.50	0.50	9
0.10	0.50	0.35	19
0.15	0.50	0.50	29
0.20	0.40	0.50	25
0.25	0.50	0.45	25
0.30	0.50	0.45	25
0.35	0.45	0.35	49
0.40	0.45	0.45	39
0.45	0.50	0.40	65
0.50	0.45	0.50	49
1.00	0.40	0.50	99
3.00	0.30	0.40	99
5.00	0.35	0.40	99

Table 9: Hyperparameters selected by cross-validation for Cod-Rna dataset.

ϵ	density	learning rate	no. estimators
0.05	0.45	0.50	5
0.10	0.50	0.40	9
0.15	0.50	0.45	9
0.20	0.50	0.40	15
0.25	0.30	0.40	9
0.30	0.35	0.50	9
0.35	0.40	0.35	15
0.40	0.45	0.40	19
0.45	0.35	0.20	19
0.50	0.45	0.25	25
1.00	0.25	0.30	29
3.00	0.20	0.20	75
5.00	0.20	0.50	29

Table 10: Hyperparameters selected by cross-validation for Mushroom dataset.

Appendix C. Gap Between Theory and Experiments for Test Error

As discussed in Section 5, there is a large gap between lower bounds predicted by large margin theory and Rademacher complexity, and the actual performance. The following table compares the best guaranteed lower bound derived by estimated Rademacher complexity and the test accuracy. The test accuracy of Adult dataset is obtained by evaluating the model on the test set, which was not touched during training. For Cod-RNA and Mushroom dataset there is no canonical test set available, so we report cross-validation accuracy.

Dataset	Rademacher Estimate of Test Accuracy	(CV) test accuracy
Adult	0.37	0.83
Cod-Rna	0.09	0.86
Mushroom	0.49	0.98

Table 11: Comparison between Rademacher estimates of generalization performance and experimental generalization performance for boosted 1-Rules, at $\epsilon = 1$.