Private Everlasting Prediction

Anonymous Author(s)
Affiliation
Address
email

Abstract

A private learner is trained on a sample of labeled points and generates a hypothesis that can be used for predicting the labels of newly sampled points while protecting the privacy of the training set [Kasiviswannathan et al., FOCS 2008]. Research uncovered that private learners may need to exhibit significantly higher sample complexity than non-private learners as is the case with, e.g., learning of one-dimensional threshold functions [Bun et al., FOCS 2015, Alon et al., STOC 2019].

We explore prediction as an alternative to learning. Instead of putting forward a hypothesis, a predictor answers a stream of classification queries. Earlier work has considered a private prediction model with just a single classification query [Dwork and Feldman, COLT 2018]. We observe that when answering a stream of queries, a predictor must modify the hypothesis it uses over time, and, furthermore, that it must use the queries for this modification, hence introducing potential privacy risks with respect to the queries themselves.

We introduce *private everlasting prediction* taking into account the privacy of both the training set *and* the (adaptively chosen) queries made to the predictor. We then present a generic construction of private everlasting predictors in the PAC model. The sample complexity of the initial training sample in our construction is quadratic (up to polylog factors) in the VC dimension of the concept class. Our construction allows prediction for all concept classes with finite VC dimension, and in particular threshold functions with constant size initial training sample, even when considered over infinite domains, whereas it is known that the sample complexity of privately learning threshold functions must grow as a function of the domain size and hence is impossible for infinite domains.

1 Introduction

A PAC learner is given labeled examples $S = \{(x_i, y_i)\}_{i\in[n]}$ drawn i.i.d. from an unknown underlying probability distribution $\mathcal{D}$ over a data domain X and outputs a hypothesis h that can be used for predicting the label of fresh points $x_{n+1}, x_{n+2}, \ldots$ sampled from the same underlying probability distribution $\mathcal{D}$ [Valiant, 1984]. It is well known that when points are labeled by a concept selected from a concept class $C = \{c : X \to \{0,1\}\}$ then learning is possible with sample complexity proportional to the VC dimension of the concept class.

Learning often happens in settings where the underlying training data is related to individuals and privacy-sensitive and where a learner is required, for legal, ethical, or other reasons, to protect personal information from being leaked in the learned hypothesis h. Private learning was introduced by Kasiviswanathan et al. [2011], as a theoretical model for

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

studying such tasks. A *private learner* is a PAC learner that preserves differential privacy with respect to its training set S. That is, the learner's distribution on outcome hypotheses must not depend too strongly on any single example in S. Kasiviswanathan et al. showed via a generic construction that any finite concept class can be learned privately and with sample complexity $n = O(\log|C|)$. This value ($O(\log|C|)$) can be significantly higher than the VC dimension of the concept class C (see below).

It is now understood that the gap between the sample complexity of private and non-private learners is essential – an important example is private learning of threshold functions (defined over an ordered domain X as $C_{thresh} = \{c_t\}_{t\in X}$ where $c_t(x) = \mathbb{1}_{x\geq t}$), which requires sample complexity that is asymptotically higher than the (constant) VC dimension of C_{thresh}. In more detail, with *pure* differential privacy, the sample complexity of private learning is characterized by the representation dimension of the concept class [Beimel et al., 2013a]. The representation dimension of C_{thresh} (hence, the sample complexity of private learning thresholds) is $\Theta(\log|X|)$ [Feldman and Xiao, 2015]. With *approximate* differential privacy, the sample complexity of learning threshold functions is $\Theta(\log^*|X|)$ [Beimel et al., 2013b, Bun et al., 2015, Alon et al., 2019, Kaplan et al., 2020, Cohen et al., 2022]. Hence, in both the pure and approximate differential privacy cases, the sample complexity grows with the cardinality of the domain $|X|$ and no private learner exists for threshold functions over infinite domains, such as the integers and the reals, whereas low sample complexity non-private learners exist for these tasks.

Privacy preserving (black-box) prediction. Dwork and Feldman [2018] proposed privacy-preserving prediction as an alternative for private learning. Noting that "[i]t is now known that for some basic learning problems [. . .] producing an accurate private model requires much more data than learning without privacy," they considered a setting where "users may be allowed to query the prediction model on their inputs only through an appropriate interface". That is, a setting where the learned hypothesis is not made public. Instead, it may be accessed in a "black-box" manner via a privacy-preserving query-answering prediction interface. The prediction interface is required to preserve the privacy of its training set S:

Definition 1.1 (private prediction interface [Dwork and Feldman, 2018] (rephrased))**.** A prediction interface M is (ϵ, δ)-differentially private if for every interactive query generating algorithm Q, the output of the interaction between Q and $M(S)$ is (ϵ, δ)-differentially private with respect to S.

Dwork and Feldman focused on the setting where the entire interaction between Q and $M(S)$ consists of issuing a single prediction query and answering it:

Definition 1.2 (Single query prediction [Dwork and Feldman, 2018])**.** Let M be an algorithm that given a set of labeled examples S and an unlabeled point x produces a label y. M is an (ϵ, δ)-differentially private prediction algorithm if for every x, the output $M(S, x)$ is (ϵ, δ)-differentially private with respect to S.

W.r.t. answering a single prediction query, Dwork and Feldman showed that the sample complexity of such predictors is proportional to the VC dimension of the concept class.

1.1 Our contributions

In this work, we extend private prediction beyond a single query to answering any sequence – *unlimited in length* – of prediction queries. We refer to this as *private everlasting prediction*. Our goal is to present a generic private everlasting predictor with low training sample complexity $|S|$.

Private prediction interfaces when applied to a large number of queries. We begin by examining private everlasting prediction under the framework of Definition 1.1. We prove:

Theorem 1.3 (informal version of Theorem 3.3)**.** *Let $\mathcal{A}$ be a private everlasting prediction interface for concept class C and assume $\mathcal{A}$ bases its predictions solely on the initial training set S, then there exists a private learner for concept class C with sample complexity $|S|$.*

This means that everlasting predictors that base their prediction solely on the initial training set S are subject to the same complexity lowerbounds as private learners. Hence, to avoid private learning lowerbounds, private everlasting predictors need to rely on more than the initial training sample S as a source of information about the underlying probability distribution and the labeling concept.

In this work, we choose to allow the everlasting predictor to rely on the queries made - which are unlabeled points from the domain X, assuming the queries are drawn from the same distribution the initial training S is sampled from. This requires changing the privacy definition, as Definition 1.1 does not protect the queries made, yet the classification given to a query can now depend on and hence reveal information provided in queries made earlier.

A definition of private everlasting predictors. Our definition of private everlasting predictors is motivated by the observations above. Consider an algorithm $\mathcal{A}$ that is first fed with a training set S of labeled points and then executes for an unlimited number of rounds, where in round i algorithm $\mathcal{A}$ receives as input a query point x_i and produces a label $\hat{y}_i$. We say that $\mathcal{A}$ is an everlasting predictor if, when the (labeled) training set S and the (unlabeled) query points are coming from the same underlying distribution, $\mathcal{A}$ answers each query points x_i with a good hypothesis h_i, and hence the label $\hat{y}_i$ produced by $\mathcal{A}$ is correct with high probability. We say that $\mathcal{A}$ is a *private* everlasting predictor if its sequence of predictions $\hat{y}_1, \hat{y}_2, \hat{y}_3, \ldots$ protects both the privacy of the training set S *and* the query points $x_1, x_2, x_3, \ldots$ in face of any adversary that adaptively chooses the query points.

We emphasize that while private everlasting predictors need to exhibit average-case utility – as good prediction is required only for the case where S and $x_1, x_2, x_3, \ldots$ are selected i.i.d. from the same underlying distribution – our privacy requirement is worst-case, and holds in face of an *adaptive* adversary that chooses each query point x_i after receiving the prediction provided for $(x_1, \ldots, x_{i-1})$, and not necessarily in accordance with any probability distribution.

A generic construction of private everlasting predictors. Our construction, called `GenericBBL`, executes in rounds. The input to the first round is the initial labeled training set S, where the number of samples in S is quadratic in the VC dimension of the concept class. Each other round begins with a collection S_i of labeled examples and ends with newly generated collection of labeled examples S_{i+1}. The set S is assumed to be consistent with some concept $c \in C$ and our construction ensures that this is the case also for the sets S_i for all i. We briefly describe the main computations performed in each round of `GenericBBL`.[1]

- **Round initialization:** At the outset of a round, the labeled set S_i is partitioned into sub-sets, each with number of samples which is proportional to the VC dimension (so we have $\approx \frac{|S_i|}{\mathrm{VC}(C)}$ sub-sets). Each of the sub-sets is used for training a classifier non-privately, hence creating a collection of classifiers $F_i = \{f : X \to \{0,1\}\}$ that are used throughout the round.
- **Query answering:** Queries are issued to the predictor in an online manner. Each query is first labeled by each of the classifiers in F_i. Then the predicted label is computed by applying a privacy-preserving majority vote on these intermediate labels. (By standard composition theorems for differential privacy, we could answer roughly $|F_i|^2 \approx \left(\frac{|S_i|}{\mathrm{VC}(C)}\right)^2$ queries without exhausting our privacy budget.) To save on the privacy budget, the majority vote is based on the `BetweenThresholds` mechanism of Bun et al. [2016] (which in turn is based on the sparse vector technique). The algorithm fails when the privacy budget is exhausted. However, when queries are sampled from the underlying distribution then with a high enough probability the labels produced by the classifiers in F_i would exhibit a clear majority.
- **Generating a labeled set for the following round:** The predictions provided in the duration of a round are not guaranteed to be consistent with any concept in C and hence cannot be used to set the following round. Instead, at the end of the round these points are relabeled consistently with C using a technique developed by Beimel et al.

[1] Important details, such as privacy amplification via sampling and management of the learning accuracy and error parameters are omitted from the description provided in this section.

[2021] in the context of private semi-supervised learning. Let S_{i+1} denote the query points obtained during the ith round, after (re)labeling them. This is a collection of size $|S_{i+1}| \approx \left(\frac{|S_i|}{\mathrm{VC}(C)}\right)^2$. Hence, provided that $|S_i| \gtrsim (\mathrm{VC}(C))^2$ we get that $|S_{i+1}| > |S_i|$ which allows us to continue to the next round with more data than we had in the previous round.

Theorem 1.4 (informal version of Theorem 5.1)**.** *For every concept class C, Algorithm* `GenericBBL` *is a private everlasting predictor requiring an initial set of labeled examples which is (upto polylogarithmic factors) quadratic in the VC dimension of C.*

1.2 Related work

Beyond the work of Dwork and Feldman [2018] on private prediction mentioned above, our work is related to private semi-supervised learning and joint differential privacy.

Semi-supervised private learning. As in the model of private semi-supervised learning of Beimel et al. [2021], our predictors depend on both labeled and unlabeled sample. Beyond the obvious difference between the models (outputting a hypothesis vs. providing black-box prediction), a major difference between the settings is that in the work of Beimel et al. [2021] all samples – labeled and unlabeled - are given at once at the outset of the learning process whereas in the setting of everlasting predictors the unlabeled samples are supplied in an online manner. Our construction of private everlasting predictors uses tools developed for the semi-supervised setting, and in particular Algorithm `LabelBoost` of of Beimel et al.

Joint differential privacy. Kearns et al. [2015] introduced joint differential privacy (JDP) as a relaxation of differential privacy applicable for mechanism design and games. For every user u, JDP requires that the outputs jointly seen by all other users would preserve differential privacy w.r.t. the input of u. Crucially, in JDP users select their inputs ahead of the computation. In our settings, the inputs to a private everlasting predictor are prediction queries which are chosen in an online manner, and hence a query can depend on previous queries and their answers. Yet, similarly to JDP, the outputs provided to queries not performed by a user u should jointly preserve differential privacy w.r.t. the query made by u. Our privacy requirement hence extends JDP to an adaptive online setting.

Additional works on private prediction. Bassily et al. [2018] studied a variant of the private prediction problem where the algorithm takes a labeled sample S and is then required to answer m prediction queries (i.e., label a sequence of m unlabeled points sampled from the same underlying distribution). They presented algorithms for this task with sample complexity $|S| \gtrsim \sqrt{m}$. This should be contrasted with our model and results, where the sample complexity is independent of m. The bounds presented by Dwork and Feldman [2018] and Bassily et al. [2018] were improved by Dagan and Feldman [2020] and by Nandi and Bassily [2020] who presented algorithms with improved dependency on the accuracy parameter in the agnostic setting.

1.3 Discussion and open problems

We show how to transform any (non-private) learner for the class C (with sample complexity proportional to the VC dimension of C) to a private everlasting predictor for C. Our construction is not polynomial time due to the use of Algorithm `LabelBoost`, and requires an initial set S of labeled examples which is quadratic in the VC dimension. We leave open the question whether $|S|$ can be reduced to be linear in the VC dimension and whether the construction can be made polynomial time. A few remarks are in order:

1. Even though our generic construction is not computationally efficient, it does result in efficient learners for several interesting special cases. Specifically, algorithm `LabelBoost` can be implemented efficiently whenever given an input sample S we could efficiently enumerate all possible dichotomies from the target class C over the points in S. In particular, this is the case for the class of 1-dim threshold functions C_{thresh}, as well as additional classes with constant VC dimension. Another notable

example is the class C^{enc}_{thresh} which intuitively is an "encrypted" version of C_{thresh}. Bun and Zhandry [2016] showed that (under plausible cryptographic assumptions) the class C^{enc}_{thresh} cannot be learned privately and efficiently, while non-private learning is possible efficiently. Our construction can be implemented efficiently for this class. This provides an example where private everlasting prediction can be done efficiently, while (standard) private learning is possible but inefficient.

2. It is now known that some learning tasks require the produced model to memorize parts of the training set in order to achieve good learning rates, which in particular disallows the learning algorithm from satisfying (standard) differential privacy [Brown et al., 2021]. Our notion of private everlasting prediction circumvents this issue, since the model is never publicly released and hence the fact that it must memorize parts of the sample is not of a direct privacy threat. In other words, our work puts forward a private learning model which, in principle, allows memorization. This could have additional applications in broader settings.
3. As we mentioned, in general, private everlasting predictors cannot base their predictions solely on the initial training set, and in this work we choose to rely on the *queries* presented to the algorithm (in addition to the training set). Our construction can be easily adapted to a setting where the content of the blackbox is updated based on *fresh unlabeled samples* (whose privacy would be preserved), instead of relying on the query points themselves. This might be beneficial to avoid poisoning attacks via the queries.

2 Preliminaries

2.1 Preliminaries from differential privacy

Definition 2.1 ((ϵ,δ)-indistinguishability)**.** Let R_0, R_1 be two random variables over the same support. We say that R_0, R_1 are (ϵ,δ)-indistinguishable if for every event E defined over the support of R_0, R_1,

$$\Pr[R_0 \in E] \leq e^{\epsilon} \cdot \Pr[R_1 \in E] + \delta \quad \text{and} \quad \Pr[R_1 \in E] \leq e^{\epsilon} \cdot \Pr[R_0 \in E] + \delta.$$

Definition 2.2. Let X be a data domain. Two datasets $x, x' \in X^n$ are called *neighboring* if $|\{i : x_i \neq x'_i\}| = 1$.

Definition 2.3 (differential privacy [Dwork et al., 2006])**.** A mechanism $M : X^n \to Y$ is (ϵ,δ)-differentially private if $M(x)$ and $M(x')$ are (ϵ,δ)-indistinguishable for all neighboring $x, x' \in X^n$.

In our analysis, we use the post-processing and composition properties of differential privacy, that we cite in their simplest form.

Proposition 2.4 (post-processing)**.** *Let $M_1 : X^n \to Y$ be an (ϵ,δ)-differentially private algorithm and $M_2 : Y \to Z$ be any algorithm. Then the algorithm that on input $x \in X^n$ outputs $M_2(M_1(x))$ is (ϵ,δ)-differentially private.*

Proposition 2.5 (composition)**.** *Let M_1 be a (ϵ_1,δ_1)-differentially private algorithm and let M_2 be (ϵ_2,δ_2)-differentially private algorithm. Then the algorithm that on input $x \in X^n$ outputs $(M_1(x), M_2(x)$ is $(\epsilon_1+\epsilon_2, \delta_1+\delta_2)$-differentially private.*

Definition 2.6 (Exponential mechanism [McSherry and Talwar, 2007])**.** Let $q : X^n \times Y \to \mathbb{R}$ be a score function defined over data domain X and output domain Y. Define $\Delta = \max(|q(x,r) - q(x',y)|)$ where the maximum is taken over all $y \in Y$ and neighbouring databases $x, x' \in X^n$. The exponential mechanism is the ϵ-differentially private mechanism which selects an output $y \in Y$ with probability proportional to $e^{\frac{\epsilon q(x,y)}{2\Delta}}$.

Claim 2.7 (Privacy amplification by sub-sampling [Kasiviswanathan et al., 2011])**.** *Let $\mathcal{A}$ be an (ε',δ')-differentially private algorithm operating on a database of size n. Let $\varepsilon \leq 1$ and let $t = \frac{n}{\varepsilon}(3 + exp(\varepsilon'))$. Construct an algorithm $\mathcal{B}$ operating the database $D = (z_i)_{i=1}^{t}$. Algorithm $\mathcal{B}$ randomly selects a subset $J \subseteq \{1,2,\ldots,t\}$ of size n, and executes $\mathcal{A}$ on $D_J = (z_i)_{i \in J}$. Then $\mathcal{B}$ is $\left(\varepsilon, \frac{4\varepsilon}{3+exp(\varepsilon')}\delta'\right)$-differentially private.*

2.2 Preliminaries from PAC learning

A concept class C over data domain X is a set of predicates $c : X \to \{0,1\}$ (called concepts) which label points of the domain X by either 0 or 1. A learner $\mathcal{A}$ for concept class C is given n examples sampled i.i.d. from an unknown probability distribution $\mathcal{D}$ over the data domain X and labeled according to an unknown target concept $c \in C$. The learner should output a hypothesis $h : X \to [0,1]$ that approximates c for the distribution $\mathcal{D}$. More formally,

Definition 2.8 (generalization error)**.** The *generalization error* of a hypothesis $h : X \to [0,1]$ with respect to concept c and distribution $\mathcal{D}$ is defined as $\mathrm{error}_{\mathcal{D}}(c,h) = \mathrm{Exp}_{x\sim\mathcal{D}}[|h(x) - c(x)|]$.

Definition 2.9 (PAC learning [Valiant, 1984])**.** Let C be a concept class over a domain X. Algorithm $\mathcal{A}$ is an (α,β,n)*-PAC learner* for C if for all $c \in C$ and all distributions $\mathcal{D}$ on X,

$$\Pr[(x_1,\ldots,x_n) \sim \mathcal{D}^n \,;\, h \sim \mathcal{A}((x_1,c(x_1)),\ldots,(x_n,c(x_n)) \,;\, \mathrm{error}_{\mathcal{D}}(c,h) \le \alpha] \ge 1-\beta,$$

where the probability is over the sampling of $(x_1,\ldots,x_n)$ from $\mathcal{D}$ and the coin tosses of $\mathcal{A}$. The parameter n is the *sample complexity* of $\mathcal{A}$.

See Appendix A for additional preliminaries on PAC learning.

2.3 Preliminaties from private learning

Definition 2.10 (private PAC learning [Kasiviswanathan et al., 2011])**.** Algorithm $\mathcal{A}$ is a $(\alpha,\beta,\epsilon,\delta,n)$-private PAC learner if (i) $\mathcal{A}$ is an (α,β,n)-PAC learner and (ii) $\mathcal{A}$ is (ϵ,δ) differentially private.

Kasiviswanathan et al. [2011] provided a generic private learner with $O(\mathrm{VC}(C)\log(|X|))$ labeled samples. Beimel et al. [2013a] introduced the representation dimension and showed that any concept class C can be privately learned with $\Theta(\mathrm{RepDim}(C))$ samples.[2] For the sample complexity of (ϵ,δ)-differentially private learning of threshold functions over domain X, Bun et al. [2015] give a lower bound of $\Omega(\log^*|X|)$. Recently, Cohen et al. [2022] give a (nearly) matching upper bound of $\tilde{O}(\log^*|X|)$.

3 Towards private everlasting prediction

In this work, we extend private prediction beyond a single query to answering any sequence – unlimited in length – of prediction queries. Our goal is to present a generic private everlasting predictor with low training sample complexity $|S|$.

Definition 3.1 (everlasting prediction)**.** Let $\mathcal{A}$ be an algorithm with the following properties:

1. Algorithm $\mathcal{A}$ receives as input n labeled examples $S = \{(x_i,y_i)\}_{i=1}^n \in (X \times \{0,1\})^n$ and selects a hypothesis $h_0 : X \to \{0,1\}$.
2. For round $r \in \mathbb{N}$, algorithm $\mathcal{A}$ gets a query, which is an unlabeled element $x_{n+r} \in X$, outputs $h_{r-1}(x_{n+r})$ and selects a hypothesis $h_r : X \to \{0,1\}$.

We say that $\mathcal{A}$ is an (α,β,n)*-everlasting predictor* for a concept class C over a domain X if the following holds for every concept $c \in C$ and for every distribution $\mathcal{D}$ over X. If $x_1,x_2,\ldots$ are sampled i.i.d. from $\mathcal{D}$, and the labels of the n initial samples S are correct, i.e., $y_i = c(x_i)$ for $i \in [n]$, then $\Pr[\exists r \ge 0 \text{ s.t. } \mathrm{error}_{\mathcal{D}}(c,h_r) > \alpha] \le \beta$, where the probability is over the sampling of $x_1,x_2,\ldots$ from $\mathcal{D}$ and the randomness of $\mathcal{A}$.

Definition 3.2. An algorithm $\mathcal{A}$ is an $(\alpha,\beta,\epsilon,\delta,n)$-everlasting differentially private prediction interface if (i) $\mathcal{A}$ is a (ϵ,δ)-differentially private prediction interface M (as in Definition 1.1), and (ii) $\mathcal{A}$ is an (α,β,n)-everlasting predictor.

As a warmup, consider an $(\alpha,\beta,\epsilon,\delta,n)$- everlasting differentially private prediction interface $\mathcal{A}$ for concept class C over (finite) domain X (as in Definition 3.2 above). Assume that $\mathcal{A}$ does not vary its hypotheses, i.e. (in the language of Definition 3.1) $h_r = h_0$ for all $r > 0$.[3] Note

[2] We omit the dependency on $\epsilon,\delta,\alpha,\beta$ in this brief review.

[3] Formally, $\mathcal{A}$ can be thought of as two mechanisms (M_0,M_1) where M_0 is (ϵ,δ)-differentially private. (i) On input a labeled training sample S mechanism M_0 computes a hypothesis h_0. (ii) On a query $x \in X$ mechanism M_1 replies $h_0(x)$.

that a computationally unlimited adversarial querying algorithm can recover the hypothesis h_0 by issuing all queries $x \in X$. Hence, in using $\mathcal{A}$ indefinitely we lose any potential benefits to sample complexity of restricting access to h_0 to being black-box and getting to the point where the lower-bounds on n from private learning apply. A consequence of this simple observation is that a private everlasting predictor cannot answer all prediction queries with a single hypothesis – it must modify its hypothesis over time as it processes new queries.

We now take this observation a step further, showing that a private everlasting predictor that answers prediction queries solely based on its training sample S is subject to the same sample complexity lowerbounds as private learners.

Consider an $(\alpha, \beta < 1/8, \epsilon, \delta, n)$-everlasting differentially private prediction interface $\mathcal{A}$ for concept class C over (finite) domain X that upon receiving the training set $S \in (X \times \{0,1\})^n$ selects an infinite sequence of hypotheses $\{h_r\}_{r\geq 0}$ where $h_r : X \to \{0,1\}$. Formally, we can think of $\mathcal{A}$ as composed of three mechanisms $\mathcal{A} = (M_0, M_1, M_2)$ where M_0 is (ϵ,δ)-differentially private:

- On input a labeled training sample $S \in (X \times \{0,1\})^n$ mechanism M_0 computes an initial state and an initial hypothesis $(\sigma_0, h_0) = M_0(S)$.
- On a query x_{n+r} mechanism M_1 produces an answer $M_1(x_{n+r}) = h_i(x_{n+r})$ and mechanism M_2 updates the hypothesis-state pair $(h_{r+1}, \sigma_{r+1}) = M_2(\sigma_r)$.

Note that as M_0 and M_2 do not receive the sequence $\{x_{n+r}\}_{r\geq 0}$ as input, the sequence $\{h_r\}_{r\geq 0}$ depends solely on S. Furthermore as M_1 and M_2 post-process the outcome of M_0, i.e., the sequence of queries and predictions $\{(x_r, h_r(x_r))\}_{r\geq 0}$ preserves (ϵ,δ)-differential privacy with respect to the training set S. In Appendix B we prove:

Theorem 3.3. *$\mathcal{A}$ can be transformed into a $(O(\alpha), O(\beta), \epsilon, \delta, O(n \log(1/\beta))$-private PAC learner for C.*

3.1 A definition of private everlasting prediction

Theorem 3.3 requires us to seek private predictors whose prediction relies on more information than what is provided by the initial labeled sample. Possibilities include requiring the input of additional labeled or unlabeled examples during the lifetime of the predictor, while protecting the privacy of these examples. In this work we choose to rely on the queries for updating the predictor's internal state. This introduces a potential privacy risk for these queries as sensitive information about a query may be leaked in the predictions following it. Furthermore, we need take into account that a privacy attacker may choose their queries adversarially and adaptively.

Definition 3.4 (private everlasting black-box prediction)**.** An algorithm $\mathcal{A}$ is an $(\alpha, \beta, \varepsilon, \delta, n)$-private everlasting black-box predictor for a concept class C if

1. **Prediction:** $\mathcal{A}$ is an (α, β, n)-everlasting predictor for C (as in Definition 3.1).
2. **Privacy:** For every adversary $\mathcal{B}$ and every $t \geq 1$, the random variables $\mathrm{View}^0_{\mathcal{B},t}$ and $\mathrm{View}^1_{\mathcal{B},t}$ (defined in Figure 1) are (ε,δ)-indistinguishable.

4 Tools from prior works

We briefly describe tools from prior works that we use in our construction. See Appendix C for a more detailed account.

Algorithm `LabelBoost` [Beimel et al., 2021]: Algorithm `LabelBoost` takes as input a partially labeled database $S \circ T \in (X \times \{0,1,\bot\})^*$ (where the first portion of the database, S, contains labeled examples) and outputs a similar database where both S and T are (re)labeled. We use the following lemmata from Beimel et al. [2021]:

Lemma 4.1 (privacy of Algorithm `LabelBoost`)**.** *Let $\mathcal{A}$ be an (ϵ,δ)-differentially private algorithm operating on labeled databases. Construct an algorithm $\mathcal{B}$ that on input a* partially

Parameters: $b \in \{0,1\}$, $t \in \mathbb{N}$.

Training Phase:

1. The adversary $\mathcal{B}$ chooses two sets of n labeled elements $(x_1^0, y_1^0), \ldots, (x_n^0, y_n^0)$ and $(x_1^1, y_1^1), \ldots, (x_n^1, y_n^1)$, subject to the restriction $\left|\left\{i \in [n] : (x_i^0, y_i^0) \neq (x_i^1, y_i^1)\right\}\right| \in \{0,1\}$.
2. If $\exists i$ s.t. $(x_i^0, y_i^0) \neq (x_i^1, y_i^1)$ then set Flag = 1. Otherwise set Flag = 0.
3. Algorithm $\mathcal{A}$ gets $(x_1^b, y_1^b), \ldots, (x_n^b, y_n^b)$ and selects a hypothesis $h_0 : X \to \{0,1\}$.
   * the adversary $\mathcal{B}$ does not get to see the hypothesis h_0 *\

Prediction phase:

4. For round $r = 1, 2, \ldots, t$:
   (a) If Flag = 1 then the adversary $\mathcal{B}$ chooses two elements $x_{n+r}^0 = x_{n+r}^1 \in X$. Otherwise, the adversary $\mathcal{B}$ chooses two elements $x_{n+r}^0, x_{n+r}^1 \in X$.
   (b) If $x_{n+r}^0 \neq x_{n+r}^1$ then Flag is set to 1.
   (c) If $x_{n+r}^0 = x_{n+r}^1$ then the adversary $\mathcal{B}$ gets $h_{r-1}(x_{n+r}^b)$.
   * the adversary $\mathcal{B}$ does not get to see the label if $x_{n+r}^0 \neq x_{n+r}^1$ *\
   (d) Algorithm $\mathcal{A}$ gets x_{n+r}^b and selects a hypothesis $h_r : X \to \{0,1\}$.
   * the adversary $\mathcal{B}$ does not get to see the hypothesis h_r *\

   Let $\mathrm{View}_{\mathcal{B},t}^b$ be $\mathcal{B}$'s entire view of the execution, i.e., the adversary's randomness and the sequence of predictions in Step 4c.

Figure 1: Definition of $\mathrm{View}_{\mathcal{B},t}^0$ and $\mathrm{View}_{\mathcal{B},t}^1$.

labeled database $S \circ T \in (X \times \{0,1,\perp\})^*$ applies $\mathcal{A}$ on the outcome of `LabelBoost`$(S \circ T)$. Then, $\mathcal{B}$ is $(\epsilon + 3, 4e\delta)$-differentially private.

Lemma 4.2 (Utility of Algorithm `LabelBoost`)**.** *Fix α and β, and let $S \circ T$ be s.t. S is labeled by some target concept $c \in C$, and s.t. $|T| \leq \frac{\beta}{e}\mathrm{VC}(C)\exp(\frac{\alpha|S|}{2\mathrm{VC}(C)}) - |S|$. Consider the execution of `LabelBoost` on $S \circ T$, and let h denote the hypothesis chosen by `LabelBoost` to relabel $S \circ T$. With probability at least $(1-\beta)$ we have that $\mathrm{error}_S(h) \leq \alpha$.*

Algorithm `BetweenThresholds` [Bun et al., 2016]: Algorithm `BetweenThresholds` takes as input a database $S \in X^n$ and thredholds t_ℓ, t_u. It applies the sparse vector technique to answer noisy threshold queries with L (below threshold) R (above threshold) and $\top$ (halt). We use the following lemmata by Bun et al. [2016] and observe that, using standard privacy amplification theorems, Algorithm `BetweenThresholds` can be modified to allow for c times of outputting $\top$ before halting, with a (roughly) $\sqrt{c}$ growth in its privacy parameter.

Lemma 4.3 (Privacy for `BetweenThresholds`)**.** *Let $\varepsilon, \delta \in (0,1)$ and $n \in \mathbb{N}$. Then algorithm `BetweenThresholds` is (ε, δ)-differentially private for any adaptively-chosen sequence of queries as long as the gap between the thresholds t_ℓ, t_u satisfies $t_u - t_\ell \geq \frac{12}{\varepsilon n}(\log(10/\varepsilon) + \log(1/\delta) + 1)$.*

Lemma 4.4 (Accuracy of `BetweenThresholds`)**.** *Let $\alpha, \beta, \varepsilon, t_\ell, t_u \in (0,1)$ and $n, k \in \mathbb{N}$ satisfy $n \geq \frac{8}{\alpha\varepsilon}(\log(k+1) + \log(1/\beta))$. Then, for any input $x \in X^n$ and any adaptively-chosen sequence of queries $q_1, q_2, \cdots, q_k$, the answers $a_1, a_2, \cdots a_{\leq k}$ produced by `BetweenThresholds` on input x satisfy the following with probability at least $1-\beta$. For any $j \in [k]$ such that a_j is returned before `BetweenThresholds` halts, (i) $a_j = \mathrm{L} \implies q_j(x) \leq t_\ell + \alpha$, (ii) $a_j = \mathrm{R} \implies q_j(x) \geq t_u - \alpha$, and (iii) $a_j = \top \implies t_\ell - \alpha \leq q_j(x) \leq t_u + \alpha$.*

Observation 1. *Using standard composition theorems for differential privacy (see, e.g., Dwork et al. [2010]), we can assume that algorithm `BetweenThresholds` takes another parameter c, and halts after c times of outputting $\top$. In this case, the algorithm satisfies $(\varepsilon', 2c\delta)$-differential privacy, for $\varepsilon' = \sqrt{2c\ln(\frac{1}{c\delta})}\varepsilon + c\varepsilon(e^\varepsilon - 1)$.*

5 A Generic Construction

Our generic construction Algorithm GenericBBL transforms a (non-private) learner for a concept class C into a private everlasting predictor for C. The proof of the following theorem follows from Theorem 5.2 and Claim 5.3 which are proved in Appendix E.

Theorem 5.1. *Given $\alpha, \beta, \delta < 1/16, \epsilon < 1$, Algorithm GenericBBL is a $(6\alpha, 4\beta, \epsilon, \delta, n)$-private everlasting predictor, where n is set as in Algorithm GenericBBL.*

Algorithm `GenericBBL`

Initial input: A labeled database $S \in (X \times \{0,1\})^n$ where $n = \frac{8\tau}{\alpha^3\varepsilon^2} \cdot \left(8\mathrm{VC}(C)\log(\frac{26}{\alpha}) + 4\log(\frac{4}{\beta})\right)^2 \cdot \log(\frac{1}{\delta}) \cdot \log^2\left(\frac{64\mathrm{VC}(C)\log(\frac{26}{\alpha})+32\log(\frac{4}{\beta})}{\varepsilon\alpha^2\beta\delta}\right) \cdot (3+\exp(\varepsilon+4))$.

1. Let $\tau > 1.1 * 10^{10}$. Set $\alpha_1 = \alpha/2$, $\beta_1 = \beta/2$. Define $\lambda_i = \frac{8\mathrm{VC}(C)\log(\frac{13}{\alpha_i})+4\log(\frac{2}{\beta_i})}{\alpha_i}$.
   /* by Theorem A.2 λ_i samples suffice for PAC learning C with parameters α_i, β_i */
2. Let $S_1 \subseteq S$ be a random subset of size $n \cdot \frac{\varepsilon}{3+\exp(\varepsilon+4)} = \frac{\tau\cdot\lambda_i^2\cdot\log(\frac{1}{\delta})\cdot\log^2(\frac{\lambda_i}{\varepsilon\alpha_i\beta_i\delta})}{\alpha_i\varepsilon}$.
3. Repeat for $i = 1, 2, 3, \ldots$
   (a) Divide S_i into $T_i = \frac{\tau\cdot\lambda_i\cdot\log(\frac{1}{\delta})\cdot\log^2(\frac{\lambda_i}{\varepsilon\alpha_i\beta_i\delta})}{\alpha_i\varepsilon}$ disjoint databases $S_{i,1}, \ldots, S_{i,T_i}$ of size λ_i.
   (b) For $t \in [T_i]$ let $f_t \in C$ be a hypothesis minimizing $\mathrm{error}_{S_{i,t}}(\cdot)$. Define $F_i = (f_1, \ldots, f_{T_i})$.
   (c) Set $R_i = \frac{25600|S_i|}{\varepsilon}$. Set $t_u = 1/2 + \alpha_i, t_\ell = 1/2 - \alpha_i$. Set the privacy parameters $\varepsilon_i' = \frac{1}{3\sqrt{c_i\ln(\frac{2}{\delta})}}$ and $\delta_i' = \frac{\delta}{2c_i}$, where $c_i = 64\alpha_i R_i$. Instantiate algorithm `BetweenThresholds` on the database of hypotheses F_i allowing for $c_i = 64\alpha_i R_i$ rounds of $\top$ while satisfying $(1,\delta)$-differential privacy (as in Observation 2).
   (d) For $\ell = 1$ to R_i:
      i. Receive as input a prediction query $x_{i,\ell} \in X$.
      ii. Give `BetweenThresholds` the query $q_{x_{i,\ell}}$ where $q_{x_{i,\ell}}(F_i) = \sum_{t\in[T_i]} f_t(x_{i,\ell})$, and obtain an outcome $y_{i,\ell} \in \{L, \top, R\}$.
      iii. Respond with the label 0 if $y_{i,\ell} = L$ and 1 if $y_{i,\ell} \in \{R, \top\}$.
      iv. If `BetweenThresholds` halts, then halt and fail (recall that `BetweenThresholds` only halts if c_i copies of $\top$ were encountered during the current iteration).
   (e) Denote $D_i = (x_{i,1}, \ldots, x_{i,R_i})$.
   (f) Let $\hat{S}_i \subseteq S_i$ and $\hat{D}_i \subseteq D_i$ be random subsets of size $\frac{\varepsilon|S_i|}{3+\exp(\varepsilon+4)}$ and $\frac{\varepsilon|D_i|}{3+\exp(\varepsilon+4)}$ respectively, and let $\hat{S}_i' \circ \hat{D}_i' \leftarrow$ `LabelBoost`$(\hat{S}_i \circ \hat{D}_i)$. Let $S_{i+1} \subseteq \hat{D}_i'$ be a random subset of size $\lambda_{i+1} T_{i+1}$.
   (g) Set $\alpha_{i+1} \leftarrow \alpha_i/2$ and $\beta_{i+1} \leftarrow \beta_i/2$.

Theorem 5.2 (accuracy of algorithm GenericBBL)**.** *Given $\alpha, \beta, \delta < 1/16$, $\varepsilon < 1$, for any concept c and any round r, algorithm GenericBBL can predict the label of x_r as $h_r(x_r)$, such that* $\Pr[error_{\mathcal{D}}(c(x_r) \neq h_r(x_r)) \leq 6\alpha] \geq 1 - 4\beta$.

Claim 5.3. *GenericBBL is (ε, δ)-differentially private.*

Remark 5.4. *For simplicity, we analyzed GenericBBL in the realizable setting, i.e., under the assumption that the training set S is* consistent *with the target class C. Our construction carries over to the agnostic setting via standard arguments (ignoring computational efficiency). We refer the reader to [Beimel et al., 2021] and [Alon et al., 2020] for generic agnostic-to-realizable reductions in the context of private learning.*

References

Noga Alon, Roi Livni, Maryanthe Malliaris, and Shay Moran. Private PAC learning implies finite littlestone dimension. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 852–860. ACM, 2019. doi: 10.1145/3313276.3316312. URL https://doi.org/10.1145/3313276.3316312.

Noga Alon, Amos Beimel, Shay Moran, and Uri Stemmer. Closure properties for private classification and online prediction. In *COLT*, volume 125 of *Proceedings of Machine Learning Research*, pages 119–152. PMLR, 2020.

Raef Bassily, Abhradeep Guha Thakurta, and Om Dipakbhai Thakkar. Model-agnostic private learning. In *NeurIPS*, pages 7102–7112, 2018.

Amos Beimel, Kobbi Nissim, and Uri Stemmer. Characterizing the sample complexity of private learners. In *ITCS*, pages 97–110. ACM, 2013a.

Amos Beimel, Kobbi Nissim, and Uri Stemmer. Private learning and sanitization: Pure vs. approximate differential privacy. In *APPROX-RANDOM*, pages 363–378, 2013b.

Amos Beimel, Kobbi Nissim, and Uri Stemmer. Learning privately with labeled and unlabeled examples. *Algorithmica*, 83(1):177–215, 2021.

Gavin Brown, Mark Bun, Vitaly Feldman, Adam D. Smith, and Kunal Talwar. When is memorization of irrelevant training data necessary for high-accuracy learning? In *STOC*, pages 123–132. ACM, 2021.

Mark Bun and Mark Zhandry. Order-revealing encryption and the hardness of private learning. In *TCC (A1)*, volume 9562 of *Lecture Notes in Computer Science*, pages 176–206. Springer, 2016.

Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil P. Vadhan. Differentially private release and learning of threshold functions. In *FOCS*, pages 634–649, 2015.

Mark Bun, Thomas Steinke, and Jonathan Ullman. Make up your mind: The price of online queries in differential privacy. *CoRR*, abs/1604.04618, 2016. URL http://arxiv.org/abs/1604.04618.

Edith Cohen, Xin Lyu, Jelani Nelson, Tamás Sarlós, and Uri Stemmer. Õptimal differentially private learning of thresholds and quasi-concave optimization. *CoRR*, abs/2211.06387, 2022. doi: 10.48550/arXiv.2211.06387. URL https://doi.org/10.48550/arXiv.2211.06387.

Yuval Dagan and Vitaly Feldman. PAC learning with stable and private predictions. In *COLT*, volume 125 of *Proceedings of Machine Learning Research*, pages 1389–1410. PMLR, 2020.

Cynthia Dwork and Vitaly Feldman. Privacy-preserving prediction. In Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet, editors, *Conference On Learning Theory, COLT 2018, Stockholm, Sweden, 6-9 July 2018*, volume 75 of *Proceedings of Machine Learning Research*, pages 1693–1702. PMLR, 2018. URL http://proceedings.mlr.press/v75/dwork18a.html.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284, 2006.

Cynthia Dwork, Guy N. Rothblum, and Salil P. Vadhan. Boosting and differential privacy. In *FOCS*, pages 51–60, 2010.

Vitaly Feldman and David Xiao. Sample complexity bounds on differentially private learning via communication complexity. *SIAM J. Comput.*, 44(6):1740–1764, 2015. doi: 10.1137/140991844. URL http://dx.doi.org/10.1137/140991844.

Haim Kaplan, Katrina Ligett, Yishay Mansour, Moni Naor, and Uri Stemmer. Privately learning thresholds: Closing the exponential gap. In *COLT*, volume 125 of *Proceedings of Machine Learning Research*, pages 2263–2285. PMLR, 2020.

Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM J. Comput.*, 40(3):793–826, 2011.

Michael J. Kearns, Mallesh M. Pai, Ryan M. Rogers, Aaron Roth, and Jonathan R. Ullman. Robust mediators in large games. *CoRR*, abs/1512.02698, 2015.

Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *FOCS*, pages 94–103. IEEE, Oct 20–23 2007.

Anupama Nandi and Raef Bassily. Privately answering classification queries in the agnostic PAC model. In *ALT*, volume 117 of *Proceedings of Machine Learning Research*, pages 687–703. PMLR, 2020.

L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, November 1984. ISSN 0001-0782. doi: 10.1145/1968.1972. URL http://doi.acm.org/10.1145/1968.1972.

Vladimir N. Vapnik and Alexey Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16 (2):264–280, 1971.

A Additional Preliminaries from PAC Learning

It is well know that that a sample of size $\Theta(\mathrm{VC}(C))$ is necessary and sufficient for the PAC learning of a concept class C, where the Vapnik-Chervonenkis (VC) dimension of a class C is defined as follows:

Definition A.1 (VC-Dimension [Vapnik and Chervonenkis, 1971])**.** Let C be a concept class over a domain X. For a set $B = \{b_1, \ldots, b_\ell\} \subseteq X$ of ℓ points, let $\Pi_C(B) = \{(c(b_1), \ldots, c(b_\ell)) : c \in C\}$ be the set of all dichotomies that are realized by C on B. We say that the set $B \subseteq X$ is *shattered* by C if C realizes all possible dichotomies over B, in which case we have $|\Pi_C(B)| = 2^{|B|}$.

The VC dimension of the class C, denoted $\mathrm{VC}(C)$, is the cardinality of the largest set $B \subseteq X$ shattered by C.

Theorem A.2 (VC bound)**.** *Let C be a concept class over a domain X. For $\alpha, \beta < 1/2$, there exists an (α, β, n)-PAC learner for C, where $n = \frac{8\mathrm{VC}(C)\log(\frac{13}{\alpha}) + 4\log(\frac{2}{\beta})}{\alpha}$.*

B Proof of Theorem 3.3

The proof of Theorem 3.3 follows from algorithms `HypothesisLearner`, `AccuracyBoost` and claims B.1, B.2, all described below.

In Algorithm `HypothesisLearner` we assume that the everlasting differentially private prediction interface $\mathcal{A}$ was fed with n i.i.d. samples taken from some (unknown) distribution $\mathcal{D}$ and labeled by an unknown concept $c \in C$. Assumning the sequence of hypotheses $\{h_r\}_{r \geq 0}$ produced by $\mathcal{A}$ satisfies

$$\forall r \;\; \mathrm{error}_{\mathcal{D}}(c, h_r) \leq \alpha \tag{1}$$

we use it to construct – with constant probability – a hypothesis h with error bounded by $O(\alpha)$.

Algorithm `HypothesisLearner`

Parameters: $0 < \beta \leq 1/8$, $R = |X|\log(|X|)\log(1/\beta)$
Input: hypothesis sequence $\{h_r\}_{r \geq 0}$

1. for all $x \in X$ let $L_x = \emptyset$
2. for $r = 0, 1, 2, \ldots, R$
   (a) select x uniformly at random from X and let $L_x = L_x \cup \{h_r(x)\}$
3. if $L_x = \emptyset$ for some $x \in X$ then fail, output an arbitrary hypothesis, and halt
   /* $\Pr[\exists x$ such that $L_x = \emptyset] \leq |X|(1 - \frac{1}{|X|})^R \approx |X|e^{-R/|X|} = \beta$ */
4. for all $x \in X$ let r_x be sampled uniformly at random from L_x
5. construct the hypothesis h, where $h(x) = r_x$

Claim B.1. *If executed on a hypothesis sequence satisfying Equation 1 then with probability at least* 3/4 *Algorithm* HypothesisLearner *outputs a hypothesis h satisfying* $\mathrm{error}_{\mathcal{D}}(c, h) \leq 8\alpha$.

Proof. Having $\mathcal{D}, c \in C$ fixed, and given a hypothesis h, we define $e_h(x)$ to be 1 if $h(x) \neq c(x)$ and 0 otherwise. Thus, we can write $\mathrm{error}_{\mathcal{D}}(c, h) = \mathbb{E}_{x \sim \mathcal{D}}[e_h(x)]$.

Observe that when Algorithm `HypothesisLearner` does not fail, r_x (and hence $h(x)$) is chosen with equal probability among $(h_1(x), h_2(x), \ldots, h_R(x))$ and hence $\mathbb{E}_\theta[e_h(x)] = \mathbb{E}_{i \in_R [R]}[e_{h_i}(x)]$ where θ denotes the randomness of `HypothesisLearner`. We get:

$$\begin{aligned}
\mathbb{E}_\theta[\mathrm{error}_{\mathcal{D}}(c, h)] &= \mathbb{E}_\theta \mathbb{E}_{x \sim \mathcal{D}}[e_h(x)] = \mathbb{E}_{x \sim \mathcal{D}} \mathbb{E}_\theta[e_h(x)] \\
&= \mathbb{E}_{x \sim \mathcal{D}} \mathbb{E}_{i \in_R [R]}[e_{h_i}(x)] = \mathbb{E}_{i \in_R [R]} \mathbb{E}_{x \sim \mathcal{D}}[e_{h_i}(x)] \\
&\leq \mathbb{E}_{i \sim \mathcal{R}}[\alpha] = \alpha.
\end{aligned}$$

By Markov inequality, we have $\Pr_\theta[\mathrm{error}_{\mathcal{D}}(c, h) \geq 8\alpha] \leq 1/8$. The claim follows noting that Algorithm `HypothesisLearner` fails with probability at most $\beta \leq 1/8$. □

The second part of the transformation is Algorithm `AccuracyBoost` that applies Algorithm `HypothesisLearner` $O(\log(1/\beta))$ times to obtain with high probability a hypothesis with $O(\alpha)$ error.

Algorithm `AccuracyBoost`

Parameters: β, $R = 104 \ln \frac{1}{\beta}$

Input: R labeled samples with n examples each $(S_1, \ldots, S_R)$ where $S_i \in (X \times \{0,1\})^n$

1. for $i = 1, 2 \ldots R$
   (a) execute $\mathcal{A}(S_i)$ to obtain a hypothesis sequence $\{h_r^i\}_{r \geq 0}$
   (b) execute Algorithm `WeakHypothesisLearner` on $\{h_r^i\}_{r \geq 0}$ to obtain hypothesis h^i
2. construct the hypothesis $\hat{h}$, where $\hat{h}(x) = \text{maj}(h^1(x), \ldots, h^R(x))$.

Claim B.2. *With probability $1 - \beta$, Algorithm `AccuracyBoost` output a 24α-good hypothesis over distribution $\mathcal{D}$.*

Proof. Define B_i to be the event where the sequence of hypotheses $\{h_r^i\}_{r \geq 0}$ produced in Step 1a of `AccuracyBoost` does not satisfy Equation 1. We have,

$$\Pr[\text{error}_{\mathcal{D}}(c, h_i) > 8\alpha] \leq \Pr[B] + (1 - \Pr[B]) \cdot \Pr[\text{error}_{\mathcal{D}}(c, h) > 8\alpha] \leq \beta + 1/4 < 3/8.$$

Hence, by the Chernoff bound, when $R \geq 104 \ln \frac{1}{\beta}$, we have at least $7R/8$ hypotheses are 8α-good over distribution $\mathcal{D}$. Consider the worst case, in which $R/8$ hypotheses always output wrong labels. To output a wrong label of x, we require at least $3R/8$ hypotheses to output wrong labels. Thus h is 24α-good over distribution $\mathcal{D}$. □

C Tools from Prior Works

C.1 Algorithm `LabelBoost` [Beimel et al., 2021]

Algorithm `LabelBoost` **[Beimel et al., 2021]**

Parameters: A concept class C.

Input: A partially labeled database $S \circ T \in (X \times \{0, 1, \perp\})^*$.

% We assume that the first portion of the database (denoted S) contains labeled examples. The algorithm outputs a similar database where both S and T are (re)labeled.

1. Initialize $H = \emptyset$.
2. Let $P = \{p_1, \ldots, p_\ell\}$ be the set of all points $p \in X$ appearing at least once in $S \circ T$. Let $\Pi_C(P) = \{(c(p_1), \ldots, c(p_\ell)) : c \in C\}$ be the set of all dichotomies generated by C on P.
3. For every $(z_1, \ldots, z_\ell) \in \Pi_C(P)$, add to H an arbitrary concept $c \in C$ s.t. $c(p_i) = z_i$ for every $1 \leq i \leq \ell$.
4. Choose $h \in H$ using the exponential mechanism with privacy parameter ϵ=1, solution set H, and the database S.
5. (Re)label $S \circ T$ using h, and denote the resulting database $(S \circ T)^h$, that is, if $S \circ T = (x_i, y_i)_{i=1}^t$ then $(S \circ T)^h = (x_i, y_i')_{i=1}^t$ where $y_i' = h(x_i)$.
6. Output $(S \circ T)^h$.

Lemma C.1 (privacy of Algorithm `LabelBoost` [Beimel et al., 2021])**.** *Let $\mathcal{A}$ be an (ϵ, δ)-differentially private algorithm operating on partially labeled databases. Construct an algorithm $\mathcal{B}$ that on input a partially labeled database $S \circ T \in (X \times \{0, 1, \perp\})^*$ applies $\mathcal{A}$ on the outcome of `LabelBoos`$(S \circ T)$. Then, $\mathcal{B}$ is $(\epsilon + 3, 4e\delta)$-differentially private.*

Consider an execution of LabelBoost on a database $S \circ T$, and assume that the examples in S are labeled by some target concept $c \in C$. Recall that for every possible labeling $\vec{z}$ of the elements in S and in T, algorithm LabelBoost adds to H a hypothesis from C that agrees with $\vec{z}$. In particular, H contains a hypothesis that agrees with the target concept c on S (and on T). That is, $\exists f \in H$ s.t. $\text{error}_S(f) = 0$. Hence, the exponential mechanism (on Step 4) chooses (w.h.p.) a hypothesis $h \in H$ s.t. $\text{error}_S(h)$ is small, provided that $|S|$ is roughly $\log|H|$, which is roughly $\text{VC}(C) \cdot \log(|S| + |T|)$ by Sauer's lemma. So, algorithm LabelBoost takes an input database where only a small portion of it is labeled, and returns a similar database in which the labeled portion grows exponentially.

Lemma C.2 (utility of Algorithm LabelBoost [Beimel et al., 2021])**.** *Fix α and β, and let $S \circ T$ be s.t. S is labeled by some target concept $c \in C$, and s.t.*

$$|T| \leq \frac{\beta}{e} \text{VC}(C) \exp\left(\frac{\alpha|S|}{2\text{VC}(C)}\right) - |S|.$$

Consider the execution of LabelBoost on $S \circ T$, and let h denote the hypothesis chosen on Step 4. With probability at least $(1-\beta)$ we have that $\text{error}_S(h) \leq \alpha$.

C.2 Algorithm BetweenThresholds [Bun et al., 2016]

Algorithm BetweenThresholds [Bun et al., 2016]

Input: Database $S \in X^n$.
Parameters: $\varepsilon, t_\ell, t_u \in (0,1)$ and $n, k \in \mathbb{N}$.

1. Sample $\mu \sim \text{Lap}(2/\varepsilon n)$ and initialize noisy thresholds $\hat{t}_\ell = t_\ell + \mu$ and $\hat{t}_u = t_u - \mu$.
2. For $j = 1, 2, \cdots, k$:
   (a) Receive query $q_j : X^n \to [0,1]$.
   (b) Set $c_j = q_j(S) + \nu_j$ where $\nu_j \sim \text{Lap}(6/\varepsilon n)$.
   (c) If $c_j < \hat{t}_\ell$, output L and continue.
   (d) If $c_j > \hat{t}_u$, output R and continue.
   (e) If $c_j \in [\hat{t}_\ell, \hat{t}_u]$, output $\top$ and halt.

Lemma C.3 (Privacy for BetweenThresholds [Bun et al., 2016])**.** *Let $\varepsilon, \delta \in (0,1)$ and $n \in \mathbb{N}$. Then algorithm BetweenThresholds is (ε, δ)-differentially private for any adaptively-chosen sequence of queries as long as the gap between the thresholds t_ℓ, t_u satisfies*

$$t_u - t_\ell \geq \frac{12}{\varepsilon n}\left(\log(10/\varepsilon) + \log(1/\delta) + 1\right).$$

Lemma C.4 (Accuracy for BetweenThresholds [Bun et al., 2016])**.** *Let $\alpha, \beta, \varepsilon, t_\ell, t_u \in (0,1)$ and $n, k \in \mathbb{N}$ satisfy*

$$n \geq \frac{8}{\alpha\varepsilon}\left(\log(k+1) + \log(1/\beta)\right).$$

Then, for any input $x \in X^n$ and any adaptively-chosen sequence of queries $q_1, q_2, \cdots, q_k$, the answers $a_1, a_2, \cdots a_{\leq k}$ produced by BetweenThresholds on input x satisfy the following with probability at least $1-\beta$. For any $j \in [k]$ such that a_j is returned before BetweenThresholds halts,

- $a_j = \text{L} \implies q_j(x) \leq t_\ell + \alpha$,
- $a_j = \text{R} \implies q_j(x) \geq t_u - \alpha$, *and*
- $a_j = \top \implies t_\ell - \alpha \leq q_j(x) \leq t_u + \alpha$.

Observation 2. *Using standard composition theorems for differential privacy (see, e.g., Dwork et al. [2010]), we can assume that algorithm BetweenThresholds takes another parameter c, and halts after c times of outputting $\top$. In this case, the algorithm satisfies $(\varepsilon', 2c\delta)$-differential privacy, for $\varepsilon' = \sqrt{2c\ln(\frac{1}{c\delta})}\varepsilon + c\varepsilon(e^\varepsilon - 1)$.*

D Some Technical Facts

We refer to the execution of steps 3a-3g of algorithm GenericBBL as a *phase* of the algorithm, indexed by $i = 1, 2, 3, \ldots$.

The original BetweenThresholds needs to halt when it outputs $\top$. In GenericBBL, we tolerance it to halt at most c_i times in the phase i. We prove BetweenThresholds in GenericBBL is $(1, \delta)$-differentially private.

Claim D.1. *For $\delta < 1$, Mechanism BetweenThresholds used in step 3c in the i-th iteration, is $(1, \delta)$-differentially private.*

Proof. Let $\varepsilon_i', \delta_i'$ be as in Step 3c. Since $e^{\varepsilon_i'} - 1 < 2\varepsilon_i'$ for $0 < \varepsilon_i' < 1$, we have

$$\sqrt{2c_i \ln(\frac{1}{c_i\delta_i'})} \cdot \varepsilon_i' + c_i\varepsilon_i'(e^{\varepsilon_i'} - 1) \le \sqrt{2c_i \ln(\frac{2}{\delta})} \cdot \varepsilon_i' + 2c_i{\varepsilon_i'}^2 = \frac{\sqrt{2}}{3} + \frac{2}{9\ln(\frac{2}{\delta})} \le 1.$$

The proof is concluded by using observation 2. □

In Claim D.2- D.5, we prove that with high probability, BetweenThresholds in step 3d halts within $64\alpha_i$ times. We prove it by 4 steps:
1.prove that with high probability, most hypothesis in step 3b have high accuracy (Claim D.2).
2.prove that if most hypothesis in step 3b have high accuracy, then with high probability, the queries in BetweenThresholds are closed to 0 or 1 (Claim D.3).
3.prove that if the queries in BetweenThresholds are closed to 0 or 1, then BetweenThresholds in step 3d will outputs L or R with high probability(Claim D.4).
4.prove that if BetweenThresholds outputs L or R, then every single phase fails with low probability(Claim D.5).

Claim D.2. *If $\beta_i \le 1/32$ and $T_i \ge 96 \ln \frac{1}{\alpha_i}$, then with probability $1 - \alpha_i$, $\frac{15T_i}{16}$ hypotheses in step 3b are α_i-good with respect to g_i, where g_i is the concept of S_i.*

Proof. By the VC bound (Theorem A.2), for each $t \in [T_i]$, we have

$$\Pr[\mathrm{error}_{\mathcal{D}}(f_t, g_i) \le \alpha_i] \ge 1 - \beta_i.$$

By Chernoff bound, if $T_i \ge \frac{16+256\beta_i}{(1-16\beta_i)^2} \ln \frac{1}{\alpha_i}$, then with probability $1 - \alpha_i$, we have $\frac{15T_i}{16}$ hypotheses have $\mathrm{error}_{\mathcal{D}}(f_t, g_i) \le \alpha_i$. When $\beta_i \le 1/32$, it is sufficient to set $T_i \ge 96 \ln \frac{1}{\alpha_i}$. □

Claim D.3. *If $\alpha_i \le 1/16$ and $\frac{15T_i}{16}$ hypotheses in step 3b are α_i-good with respect to g_i, where g_i is the concept of S_i, then $\Pr_{x\sim\mathcal{D}}[|q(x) - \frac{1}{2}| \le \frac{3}{8}] \le 15\alpha_i$.*

Proof. W.l.o.g. assume $g_i(x) = 1$, where g_i is the concept of S_i, so it is sufficient to prove $\Pr_{x\sim\mathcal{D}}[q(x) \le \frac{7}{8}] \le 8\alpha_i$. Consider the worst case that $\frac{T_i}{16}$ "bad" hypotheses output 0. In that case, $q(x) \le \frac{7}{8}$ when $\frac{T_i}{16}$ of α_i-good hypotheses output 0. So that with probability $15\alpha_i$, we have $q(x) \le \frac{7}{8}$.(see Figure 2)

□

Claim D.4. *Let $t_u < 1/2 + 1/8$ and $t_\ell > 1/2 - 1/8$. For a query q such that $q(S) > 7/8$ (similarly, for $q(S) < 1/8$), Algorithm BetweenThresholds outputs R (similarly, L) with probability at least $1 - exp\left(-\frac{T_i}{144\sqrt{c_i \ln(\frac{2}{\delta})}}\right)$.*

Proof. Wlog assume $q(S) > 7/8$, it is sufficient to show

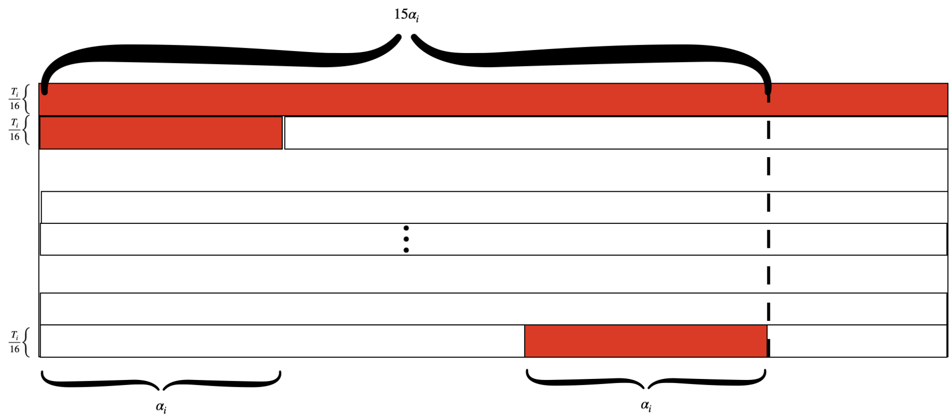

Figure 2: The horizontal represents the input point. The vertical represents the hypothesis. The red parts represent the incorrect prediction. We let $\frac{T_i}{16}$ hypothesis predict all labels as 0. To let $q(x) \le \frac{7}{8}$, there must exist $\frac{T_i}{16}$ hypothesis output 0. In the worst case, at most $15\alpha_i$ of points are labeled as 0.

$$\begin{aligned}
\Pr[\texttt{BetweenThreshold}\ \text{outputs} R] &= \Pr[q(S) + \mathrm{Lap}(6/\varepsilon' T_i) > t_u + \mathrm{Lap}(2/\varepsilon' T_i)] \\
&> \Pr[\mathrm{Lap}(6/\varepsilon' T_i) > -1/8] \cdot \Pr[\mathrm{Lap}(2/\varepsilon' T_i) < 1/8] \\
&= \left(1 - \frac{1}{2}\exp\left(-\frac{T_i}{144\sqrt{c_i \ln(\frac{2}{\delta})}}\right)\right) \cdot \left(1 - \frac{1}{2}\exp\left(-\frac{T_i}{48\sqrt{c_i \ln(\frac{2}{\delta})}}\right)\right) \\
&> 1 - \exp\left(-\frac{T_i}{144\sqrt{c_i \ln(\frac{2}{\delta})}}\right).
\end{aligned}$$

□

Claim D.5. *For any phase i, `BetweenThresholds` outputs $\top$ at most $64\alpha_i R_i$ times with probability at most β_i.*

Proof. For a single query, if $t_u < 1/2 + 1/8$ and $q(S) > 7/8$ (similarly, $t_\ell > 1/2 - 1/8$ and $q(S) < 1/8$), by Claim D.4, `BetweenThresholds` outputs $\top$ with probability at most $\exp\left(-\frac{T_i}{144\sqrt{c_i \ln(\frac{2}{\delta})}}\right) = \exp\left(-\frac{T_i}{144\sqrt{64\alpha_i R_i \ln(\frac{2}{\delta})}}\right) < \alpha_i$. Combine Claim D.2 and D.3, `BetweenThresholds` outputs $\top$ with probability at most $32\alpha_i$. By the Chernoff bound and $R_i \ge \frac{3\ln(\frac{1}{\beta_i})}{\alpha_i}$, `BetweenThresholds` outputs $\top$ more than $64\alpha_i R_i$ times with probability at most β_i. □

In step 3f, `GenericBBL` takes a random subset of size $\lambda_{i+1} T_{t+1}$ from $\hat{D}'_i$. We show that the size of $\hat{D}'_i$ is at least $\lambda_{i+1} T_{t+1}$.

Claim D.6. *When $\varepsilon \le 1$, for any $i \ge 1$, we always have $|\hat{D}'_i| \ge \lambda_{i+1} T_{i+1}$.*

Proof. Let $m = 3 + \exp(\varepsilon + 4) < 200$. By the step 3c, step 3e and step 3f, $|\hat{D}_j| = \frac{\varepsilon |D_j|}{m} = \frac{25600|S_j|}{m} \ge 128|S_j| = 128\lambda_j T_j$. Then it is sufficient to verify $128\lambda_j T_j \ge \lambda_{j+1} T_{j+1}$

We can verify that

$$4\lambda_j = 4\cdot\frac{8\mathrm{VC}(C)\log(\frac{13}{\alpha_i}) + 4\log(\frac{2}{\beta_i})}{\alpha_i} = 4\cdot\frac{8\mathrm{VC}(C)(\log(\frac{13}{\alpha_{j+1}})-1) + 4(\log(\frac{2}{\beta_{j+1}})-1)}{2\alpha_{j+1}} \geq \lambda_{j+1}$$

and

$$32T_j = \frac{32\tau\cdot\lambda_i\cdot\log(\frac{1}{\delta})\cdot\log^2(\frac{\lambda_i}{\varepsilon\alpha_i\beta_i\delta})}{\alpha_i\varepsilon} \geq \frac{32\tau\cdot\lambda_i\cdot\log(\frac{1}{\delta})\cdot\log^2(\frac{\lambda_{i+1}}{16\varepsilon\alpha_{i+1}\beta_{i+1}\delta})}{8\alpha_{i+1}\varepsilon} \geq \lambda_{j+1}T_{j+1}.$$

The last inequalitu holds because $\lambda_j \geq 4$ and $\alpha_j, \beta_j \leq 1/2$. □

To apply the privacy and accuracy of *LabelBoost* and *BetweenThresholds*, the sizes of the databases need to satisfy the inequalities in lemma C.2, C.3 and C.4. We verify that in each phase, the sizes of the databases always satisfy the requirement.

Claim D.7. *Let $\alpha, \beta, \delta < 1/16$, $\varepsilon \leq 1$, and $\mathrm{VC}(C) \geq 1$. Then for any $i \geq 1$, we have*

$$T_i \geq \frac{8}{\alpha_i\varepsilon'}(\log(|D_i|+1) + \log(1/\beta_i)).$$

Proof. By claim D.6 and step 3c, $|D_i| = \frac{25600|S_i|}{\varepsilon} = \frac{25600\lambda_i T_i}{\varepsilon}$. Since

$$\begin{aligned}
\frac{8}{\alpha_i\varepsilon'}(\log(|D_i|+1) + \log(1/\beta_i)) &= \frac{24\sqrt{64\alpha_i|D_i|\ln(\frac{2}{\delta})}}{\sqrt{2}\alpha_i}\cdot(\log(|D_i|+1) + \log(1/\beta_i)) \\
&= O\left(\sqrt{\frac{\lambda_i T_i \log(\frac{1}{\delta})}{\alpha_i\varepsilon}}\left(\log(\frac{\lambda_i T_i}{\varepsilon\beta_i})\right)\right) \\
&= O\left(\sqrt{\frac{\lambda_i T_i \log(\frac{1}{\delta})}{\alpha_i\varepsilon}}\cdot\log\left(\frac{\lambda_i\log(\frac{1}{\delta})}{\alpha_i\beta_i\varepsilon}\right)\right),
\end{aligned}$$

and $T_i = \frac{\tau\cdot\lambda_i\cdot\log(\frac{1}{\delta})\cdot\log^2(\frac{\lambda_i}{\varepsilon\alpha_i\beta_i\delta})}{\alpha_i\varepsilon}$, where $\tau \geq 1.1 * 10^{10}$, the inequality always holds. □

Claim D.8. *When $\varepsilon \leq 1$, for any $i \geq 1$, we have $|\hat{D}_i| \leq \frac{\beta_i}{e}\mathrm{VC}(C) exp\left(\frac{\alpha_i|\hat{S}_i|}{2\mathrm{VC}(C)}\right) - |\hat{S}_i|$.*

Proof. By claim D.6, step 3c and step 3f,

$$|\hat{D}_i| = \frac{\varepsilon|D_i|}{m} = O(\lambda_i T_i) = O\left(\mathrm{VC}(C)\log^2(\mathrm{VC}(C))\cdot\mathrm{poly}\left(\frac{1}{\alpha_i}, \log(\frac{1}{\beta_i}), \frac{1}{\varepsilon}, \log(\frac{1}{\delta})\right)\right)$$

and

$$\begin{aligned}
|\hat{S}_i| &= \frac{\varepsilon|S_i|}{m} \\
&= O(\varepsilon\lambda_i T_i) = O(\lambda_i T_i) \\
&= O\left(\mathrm{VC}(C)\log^2(\mathrm{VC}(C))\cdot\mathrm{poly}\left(\frac{1}{\alpha_i}, \log(\frac{1}{\beta_i}), \frac{1}{\varepsilon}, \log(\frac{1}{\delta})\right)\right).
\end{aligned} \tag{2}$$

Note that

$$\frac{\beta_i}{e}\mathrm{VC}(C)\exp\left(\frac{\alpha_i|\hat{S}_i|}{2\mathrm{VC}(C)}\right) = \Omega\left(\mathrm{VC}^2(C)\cdot\exp\left(\mathrm{poly}\left(\frac{1}{\alpha_i}, \log(\frac{1}{\beta_i}), \frac{1}{\varepsilon}, \log(\frac{1}{\delta})\right)\right)\right),$$

for $T_i = \frac{\tau\cdot\lambda_i\cdot\log(\frac{1}{\delta})\cdot\log^2(\frac{\lambda_i}{\varepsilon\alpha_i\beta_i\delta})}{\alpha_i\varepsilon}$, the inequality holds when $\tau \geq 1$. □

Claim D.9. *For every $i \ge 1$, we have*
$$t_u - t_\ell \ge \frac{12}{\varepsilon_i' T_i}\left(\log(10/\varepsilon_i') + \log(1/\delta_i') + 1\right).$$

Proof. By step 3c, $t_u - t_\ell = 2\alpha_i$. Then we have
$$\begin{aligned}
\frac{6}{\alpha_i \varepsilon_i' T_i}\left(\log(10/\varepsilon_i') + \log(1/\delta_i') + 1\right) &= \frac{6\sqrt{64\alpha_i R_i \ln(\frac{2}{\delta})}}{\alpha_i T_i}\left(\log(10/\varepsilon_i') + \log(1/\delta_i') + 1\right)\\
&= 6\sqrt{\frac{1638400\ln(\frac{2}{\delta})\lambda_i}{\alpha_i T_i}}\left(\log(10/\varepsilon_i') + \log(1/\delta_i') + 1\right)\\
&= 6\sqrt{\frac{1638400\ln(\frac{2}{\delta})}{\tau\log(\frac{1}{\delta})\log^2(\frac{\lambda_i}{\varepsilon\alpha_i\beta_i\delta})}}\left(\log(10/\varepsilon_i') + \log(1/\delta_i') + 1\right)\\
&= O(1),
\end{aligned}$$
the inequality holds when $\tau > 10^{10}$. □

E Accuracy of Algorithm `GenericBBL` – proof of Theorem 5.2

We refer to the execution of steps 3a-3g of algorithm `GenericBBL` as a *phase* of the algorithm, indexed by $i = 1, 2, 3, \ldots$.

We give some technical facts in Appendix D. In Claim E.1, we show that in each phase, samples are labeled with high accuracy. In Claim E.2, we prove that algorithm `GenericBBL` fails with low probability. In Claim E.4, we prove that algorithm `GenericBBL` predict the labels with high accuracy.

Claim E.1. *When Algorithm `GenericBBL` does not fail on phases* 1 *to i, then for phase $i+1$ we have*
$$\Pr\left[\exists g_{i+1} \in C \text{ s.t. } \mathrm{error}_{S_{i+1}}(g_{i+1}) = 0 \text{ and } \mathrm{error}_{\mathcal{D}}(g_{i+1}, c) \le \sum_{j=1}^{i+1}\alpha_j\right] \ge 1 - 2\sum_{j=0}^{i+1}\beta_j.$$

Proof. The proof is by induction on i. The base case for $i = 1$ is trivial, with $g_1 = c$. Assume the claim holds for all $j \le i$. By the properties of `LabelBoost` (Lemma C.2) and Claim D.8, with probability at least $1 - \beta_{i+1}$ we have that S_{i+1} is labeled by a hypothesis $g_{i+1} \in C$ s.t. $\mathrm{error}_{S_i}(g_i, g_{i+1}) \le \alpha_{i+1}$. Observe that the points in S_i (without their labels) are chosen i.i.d. from $\mathcal{D}$, and hence, By Theorem A.2 (VC bounds) and $|S_i| \ge 128\lambda_i \ge \lambda_{i+1}$, with probability at least $1 - \beta_{i+1}$ we have that $\mathrm{error}_{\mathcal{D}}(g_i, g_{i+1}) \le \alpha_{i+1}$. Hence, with probability $1 - 2\beta_{i+1}$, we have $\mathrm{error}_{\mathcal{D}}(g_i, g_{i+1}) \le \alpha_{i+1}$. Finally, by the triangle inequality, $\mathrm{error}_{\mathcal{D}}(g_{i+1}, c) \le \sum_{j=1}^{i+1}\alpha_j$, except with probability $2\sum_{j=1}^{i+1}\beta_j$ □

Define the following good event.

> **Event E_1:** Algorithm `GenericBBL` never fails on the execution of `BetweenThresholds` in step 3(d)iv.

Claim E.2. *Event E_1 occurs with probability at least $1 - \beta$.*

Proof. Using to union bound and Claim D.5,
$$\Pr[\text{Event } E_1 \text{ occurs}] \ge 1 - \beta.$$
□

Combining claims E.1 and E.2, we get:

Claim E.3. *Let $\mathcal{D}$ be an underlying distribution and let $c \in C$ be a target concept. Then*
$$\Pr[\forall i\, \exists g_i \in C \text{ s.t. } \mathrm{error}_{S_i}(g_i) = 0 \text{ and } \mathrm{error}_{\mathcal{D}}(g_i, c) \le \alpha] \ge 1 - 3\beta.$$

Notations. Consider the ith phase of Algorithm `GenericBBL`, and focus on the j-th iteration of Step 3. Fix all of the randomness in `BetweenThresholds`. Now observe that the output on step 3(d)iii is a deterministic function of the input $x_{i,j}$. This defines a hypothesis which we denote as $h_{i,j}$.

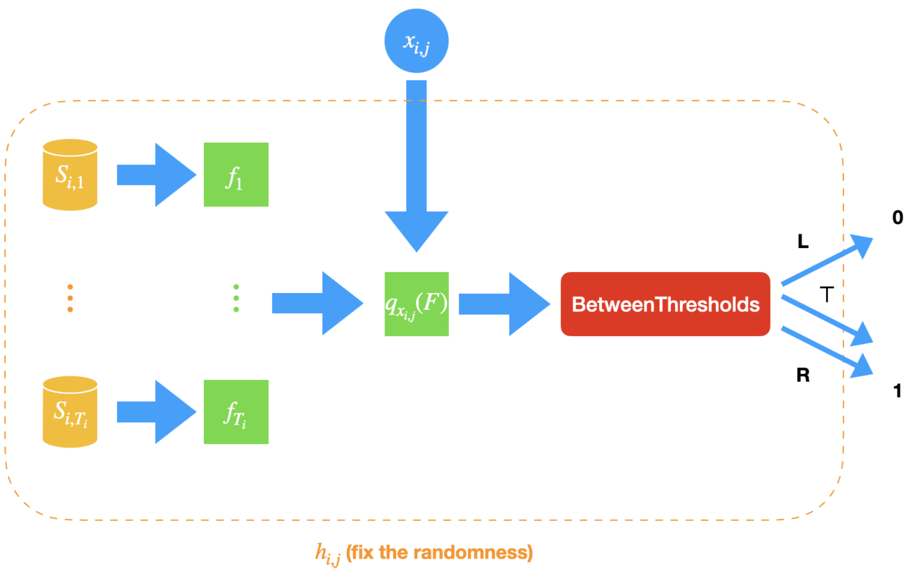

Figure 3: Hypothesis $h_{i,j}$

Claim E.4. *For $\beta < 1/16$, with probability at least $1-4\beta$, all of the hypotheses defined above are 6α-good w.r.t. $\mathcal{D}$ and c.*

Proof. In the phase i, by Claim E.3, with probability at least $1-3\beta$ we have that S_i is labeled by a hypothesis $g_i \in C$ satisfying $\text{error}_{\mathcal{D}}(g_i, c) \le \alpha$. We continue with the analysis assuming that this is the case.

On step 3a of the ith phase we divide S_i into T_i subsamples of size λ_i each, identify a consistent hypothesis $f_t \in C$ for every subsample $S_{i,t}$, and denote $F_i = (f_1, \ldots, f_T)$. By Theorem A.2 (VC bounds), every hypothesis in F_i satisfies $\text{error}_{\mathcal{D}}(f_t, g_i) \le \alpha$ with probability $3/4$, in which case, by the triangle inequality we have that $\text{error}_{\mathcal{D}}(f_t, c) \le 2\alpha$.

Set $T_i \ge \frac{512(1-4\beta_i)\ln(\frac{1}{\beta_i})}{(1-64\beta_i)^2}$, using Chernoff bound, it holds that for at least $15T_i/16$ of the hypotheses in F_i have $\text{error}_{\mathcal{D}}(f_t, g_i) \le 2\alpha$ with probability at least $1-\beta_i$. These hypotheses have $\text{error}_{\mathcal{D}}(f_t, c) \le 3\alpha$.

Let $m : X \to \{0,1\}$ defined as $m(x) = \text{maj}_{f_t \in F_i}(f_t(x))$. For m to err on a point x (w.r.t. the target concept c), it must be that at least $7/16$-fraction of the 3α-good hypotheses in $\hat{F}_i$ err on x. Consider the worst case in Figure 4 , we have $\text{error}_{\mathcal{D}}(m, c) \le 6\alpha$

By Lemma C.4 and Claim D.7, with probability at least $1-\beta_i$, all of the hypotheses defined during the ith iteration satisfy this condition, and are hence 6α-good w.r.t. c and $\mathcal{D}$. By the union bound, with probability $1-4\beta$, all the hypotheses are 6α-good. □

E.1 Privacy analysis – proof of Claim 5.3

Fix $t \in \mathbb{N}$ and the adversary $\mathcal{B}$. We need to show that $\text{View}^0_{\mathcal{B},t}$ and $\text{View}^1_{\mathcal{B},t}$ (defined in Figure 1) are $(\varepsilon, \delta)-$*indistinguishable*. We will consider separately the case where the executions differ in the training phase (Claim E.5) and the case where the difference occurs during the prediction phase (Claim E.6).

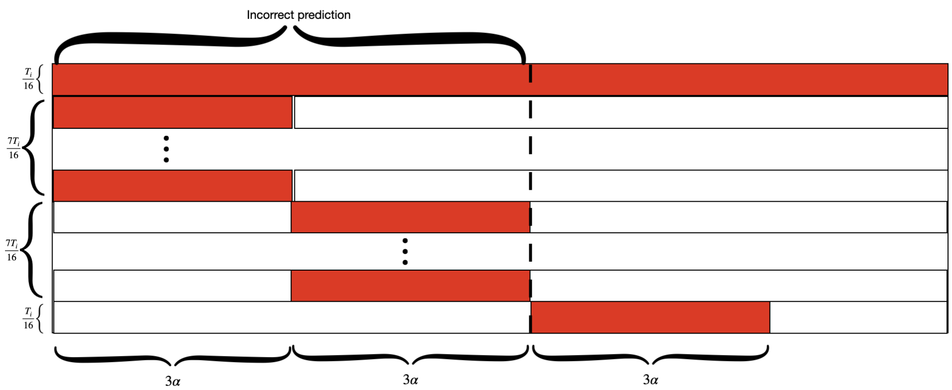

Figure 4: The horizontal represents the input point. The vertical represents the hypothesis. The red parts represent the incorrect prediction. We let $\frac{T_i}{16}$ hypothesis predict all labels incorrectly. To output an incorrect label, there must exist $\frac{7T_i}{16}$ hypothesis output the incorrect label. In the worst case, at most 6α of points are incorrectly classified.

Privacy of the initial training set S**.** Let $S^0, S^1 \in (X \times \{0,1\})^n$ be neighboring datasets of labeled examples and let $\text{View}^0_{\mathcal{B},t}$ and $\text{View}^1_{\mathcal{B},t}$ be as in Figure 1 where $\left((x^0_1, y^0_1), \ldots, (x^0_n, y^0_n)\right) = S^0$ and $\left((x^1_1, y^1_1), \ldots, (x^1_n, y^1_n)\right) = S^1$.

Claim E.5. *For all adversaries* $\mathcal{B}$*, for all* $t > 0$*, and for any two neighbouring database* S^0 *and* S^1 *selected by* $\mathcal{B}$*,* $\text{View}^0_{\mathcal{B},t}$ *and* $\text{View}^1_{\mathcal{B},t}$ *are* (ε, δ)*-indistinguishable.*

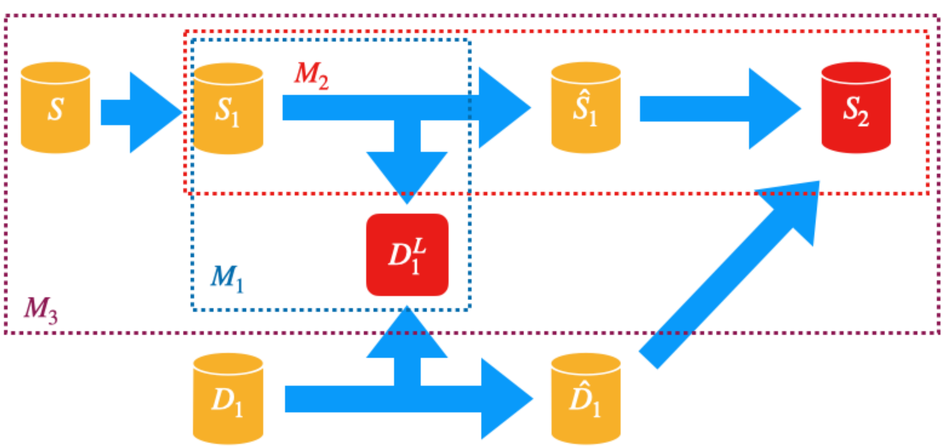

Figure 5: Privacy of the labeled sample S

Proof. Let $R'_1 = \min(t, R_1)$. Note that $\text{View}^b_{\mathcal{B},R'_1}$ is a prefix of $\text{View}^b_{\mathcal{B},t}$ which includes the labels Algorithm `GenericBBL` produces in Step 3(d)iii for the R'_1 first unlabeled points selected by $\mathcal{B}$. Let S^b_2 be the result of the first application of algorithm `LabelBoost` in Step 3f of `GenericBBL` (if $t < R_1$ we set S^b_2 as $\perp$). The creation of these random variables is depicted in Figure 5, where D^L_1 denotes the labels Algorithm `GenericBBL` produces for the unlabeled points D_1.

Observe that $\text{View}^b_{\mathcal{B},t}$ results from a post-processing (jointly by the adversary $\mathcal{B}$ and Algorithm `GenericBBL`) of the random variable $\left(\text{View}^b_{\mathcal{B},R'_1}, S^b_2\right)$, and hence it suffices to show that $\left(\text{View}^0_{\mathcal{B},R'_1}, S^0_2\right)$ and $\left(\text{View}^1_{\mathcal{B},R'_1}, S^1_2\right)$ are (ε,δ)-indistinguishable.

We follow the processes creating $\text{View}^b_{\mathcal{B},t}$ and S^b_2 in Figure 5: (i) The mechanism M_1 corresponds to the loop in Step 3d of `GenericBBL` where labels are produced for the adversarially chosen points D^b_1. By application of Lemma C.3, M_1 is $(1,\delta)$-differentially private. (ii) The mechanism M_2, corresponds to the subsampling of $\hat{S}^b_1$ from S^b_1 and the application of procedure `LabelBoost` on the subsample in Step 3f of `GenericBBL` resulting in S^b_2. By application of Claim 2.7 and Lemma C.1, M_2 is $(\varepsilon,0)$-differentially private. Thus (M_1,M_2) is $(\varepsilon+1,\delta)$-differentially private. (iii) The mechanism M_3 with input of S^b and output $\left(D^{b,L}_1, S^b_2\right) = \left(\text{View}^b_{\mathcal{B},R'_1}, S^b_2\right)$ applies (M_1,M_2) on the sub-sample S^b_1 obtained from S^b in Step 2 of `GenericBBL`. By application of Claim 2.7 M_3 is $(\varepsilon, \frac{4\varepsilon\delta}{3+\exp(\varepsilon+1)})$-differentially private. Since $\frac{4\varepsilon\delta}{3+\exp(\varepsilon+1)} \le \delta$ for any ε, hence $\left(\text{View}^0_{\mathcal{B},R'_1}, S^0_2\right)$ and $\left(\text{View}^1_{\mathcal{B},R'_1}, S^1_2\right)$ are (ε,δ)-indistinguishable □

Privacy of the unlabeled points D. Let $D^0, D^1 \in X^t$ be neighboring datasets of unlabeled examples and let $\text{View}^0_{\mathcal{B},t}$ and $\text{View}^1_{\mathcal{B},t}$ be as in Figure 1 where $\left(x^0_1,\ldots,x^0_t\right) = D^0$ and $\left(x^1_1,\ldots,x^1_t\right) = D^1$.

Claim E.6. *For all adversaries $\mathcal{B}$, for all $t > 0$, and for any two neighbouring databases D^0 and D^1 selected by $\mathcal{B}$, $\text{View}^0_{\mathcal{B},t}$ and $\text{View}^1_{\mathcal{B},t}$ are (ε,δ)-indistinguishable.*

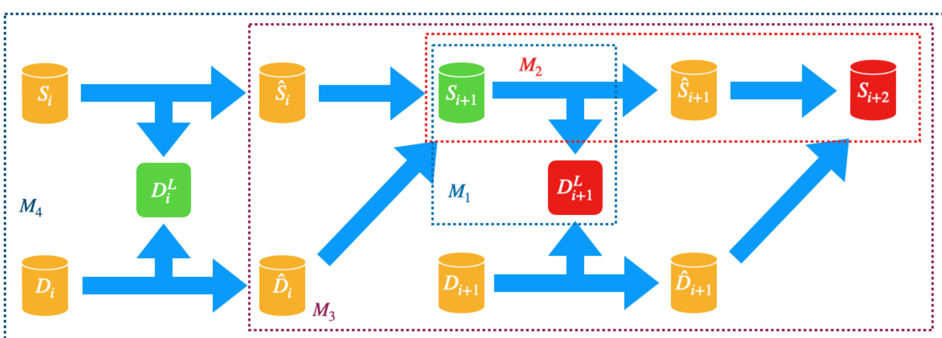

Figure 6: Privacy leakage of D_i

Proof. Let $D^0_1, D^0_2, \ldots, D^0_k$ and $D^1_1, D^1_2, \ldots, D^1_k$ be the set of unlabeled databases in step 3e of GenericBBL. Without loss of generality, we assume D^0_i and D^1_i differ on one entry. When $i = k$, $\text{View}^0_{\mathcal{B},t} = \text{View}^1_{\mathcal{B},t}$ because all selected hypothesis are the same. When $i < k$, let $R' = \min\left(\sum_{j=1}^{i+1} R_j, t\right)$.

Similar to the analysis if Claim E.5, $\text{View}^b_{\mathcal{B},t}$ results from a post-processing of the random variable $(\text{View}^b_{\mathcal{B},R'}, S^b_{i+2})$ (if $t < \sum_{j=1}^{i+1} R_j$ we set S^b_{i+2} as $\perp$). Note that $\text{View}^b_{\mathcal{B},R'_1} = (D^{b,L}_1, \ldots, D^{b,L*}_i, D^{b,L}_{i+1})$, and $(D^{b,L}_1, \ldots, D^{b,L}_{i-1}, D^{b,L*}_i)$ follow the same distribution for $b \in \{0,1\}$, where $D^{b,L*}_i$ is the labels of points in D^b_i expect the different point. So that it suffices to show that $\left(D^{0,L}_{i+1}, S^0_2\right)$ and $\left(D^{1,L}_{i+1}, S^1_2\right)$ are (ε,δ)-indistinguishable.

We follow the processes creating $D^{b,L}_{i+1}$ and S^b_{i+2} in Figure 6: (i) The mechanism M_1 corresponds to the loop in Step 3d of `GenericBBL` where labels are produced for the adversarially chosen points D^b_{i+1}. By application of Lemma C.3, M_1 is $(1, \delta)$-differentially private. (ii) The mechanism M_2, corresponds to the subsampling of $\hat{S}^b_{i+1}$ from S^b_{i+1} and the application of procedure `LabelBoost` on the subsample in Step 3f of `GenericBBL` resulting in S^b_{i+2}. By application of Claim 2.7 and Lemma C.1, M_2 is $(\varepsilon, 0)$-differentially private. Thus (M_1, M_2) is $(\varepsilon + 1, \delta)$-differentially private. (iii) The mechanism M_3 with input of $\hat{D}^b_i$ and output $\left(D^{b,L}_{i+1}, S^b_{i+2}\right)$ applies (M_2, M_3) on S_{i+1}, which is generated from $\hat{D}^b_i$ and in Step 3f of `GenericBBL`. By application of Claim C.1, M_3 is $(\varepsilon + 4, 4\varepsilon\delta)$-differentially private. (iv) The mechanism M_4, corresponds to the subsampling $\hat{D}^b_i$ from D^b_i and the application of M_4 on $\hat{D}^b_i$. By application of Claim 2.7, M_4 is $(\varepsilon, \frac{16e\varepsilon\delta}{3+\exp(\varepsilon+4)})$-differentially private. Since $\frac{16e\varepsilon}{3+\exp(\varepsilon+4)} \leq 1$ for any ε, $\left(D^{0,L}_{i+1}, S^0_2\right)$ and $\left(D^{1,L}_{i+1}, S^1_2\right)$ are (ε, δ)-indistinguishable. □

Remark E.7. *The above proofs work on the adversarially selected D because: (i) Lemma C.3 works on the adaptively selected queries. (We treat the hypothesis class F_i as the database, the unlabelled points $x_{i,\ell}$ as the query parameters.) (ii) `LabelBoost` generates labels by applying one private hypothesis on points. The labels are differentially private by post-processing.*