Differentially Private Learning with Kernels

Prateek Jain PRAJAIN@MICROSOFT.COM
Microsoft Research Labs, Bangalore, INDIA

Abhradeep Thakurta B-ABHRAG@MICROSOFT.COM
Stanford University and Microsoft Research Silicon Valley Campus

Abstract

In this paper, we consider the problem of differentially private learning where access to the training features is through a kernel function only. As mentioned in Chaudhuri et al. (2011), the problem seems to be intractable for general kernel functions in the standard learning model of releasing different private predictor. We study this problem in three simpler but practical settings. We first study an interactive model where the user sends its test points to a *trusted learner* (like search engines) and expects accurate but differentially private predictions. In the second model, the learner has access to a subset of the unlabeled test set using which it releases a predictor, which preserves privacy of the training data. (NIH, 2003) is an example of such publicly available test set. Our third model is similar to the traditional model, where learner is oblivious to the test set but the kernels are restricted to functions over vector spaces. For each of the models, we derive differentially private learning algorithms with provable "utility" or error bounds. Moreover, we show that our methods can also be applied to the traditional model where they demonstrate better dimensionality dependence when compared to the methods of (Rubinstein et al., 2009; Chaudhuri et al., 2011). Finally, we provide experimental validation of our methods.

1. Introduction

Modern systems can log and mine a lot of data to learn interesting trends/patterns which can then be used for different tasks. Typically, these logs contain sensitive information from individuals and releasing the learned patterns/model might compromise an individual's privacy. To address this issue, there have been several works that study the problem of privacy preserving learning (Rubinstein et al., 2009; Pathak et al., 2010; Chaudhuri et al., 2011).

Typically, the goal of these works is to learn a privacy preserving parameter vector $\mathbf{w}^*$ (for example a linear classifier) from the data which also generalizes well on unseen test data. There are several notions of privacy in the literature, but differential privacy (Dwork et al., 2006b) has been one of the most theoretically sound and widespread notion.

In particular, Chaudhuri et al. (2011); Rubinstein et al. (2009) introduced a general framework for differentially private regularized empirical risk minimization (ERM) that guarantees privacy as well as small "excess" error, in addition to the generalization error of the best non-private solution. Popular applications of such methods include private version of Support Vector Machines (SVM) and logistic regression.

However, the above private algorithms are mostly restricted to the "linear" case only, where each data point lies in a low dimensional vector space and can be explicitly accessed. Moreover, to provide privacy, these methods need to add noise proportional to the dimensionality of the data (d). Hence, their generalization error guarantees become weak for high-dimensional datasets. Chaudhuri et al. (2011) and Rubinstein et al. (2009) briefly looked at kernel ERMS where the dimensionality (d) can be allowed to be potentially infinite. However, their algorithms are restricted to the class of translation invariant kernels and does not capture a wide variety of kernels, e.g., polynomial kernels or pyramid match kernel from the computer vision domain (Grauman & Darrell, 2007).

In this work, we study the problem of differentially private learning using kernel ERM (kERM), where access to each data point is through a kernel function only. As the optimal solution to kERMs can only

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28. Copyright 2013 by the author(s).

be obtained implicitly and is defined in terms of the training data points themselves, it is not clear how a solution to general kERMs can be released privately (Chaudhuri et al., 2011). To address this issue, we propose three simpler but practical models for privacy preserving kERM. For each model, we provide an algorithm that guarantees differential privacy to the training points, while also providing strong "utility" guarantees, i.e., error bounds. In contrast to bounds by (Chaudhuri et al., 2011; Rubinstein et al., 2009), our bounds are independent of the dimensionality of the feature space.

The key insight behind our models is that the end goal of solving kERM for a user is to obtain accurate predictions for its test set. Now, while providing explicit access to the optimum of the kERM (without violating privacy) might be difficult, privacy preserving *predictions* on the test set should still be possible. Here, by "privacy preserving" we mean privacy of the training data points. Each of our model tries to exploit this insight in a different setting.

Our first model (see Figure 1 (a)) is an interactive model, where the user sends its test data to a trusted learner (who solves an ERM over the training points), and the learner sends back predictions over those points. A typical scenario would be an online ad system where the learner (for example, Google) uses click-logs to predict whether a particular ad is relevant for a given user. See Section 5 for more details. For this model, we adapt an algorithm by Gupta et al. (2011) and show that we can provide accurate privacy preserving predictions for a large number (exponentially many in the training set size) of test points.

Our second model is "semi-interactive", where the user sends a small subset of its test data to the trusted learner and the learner returns a differentially private version $\widehat{\mathbf{w}}$ of the optimum to the kERM ($\mathbf{w}^*$). An example scenario can be a learner using a private dataset from a biology lab to learn a model that it uses to predict a disease over some public dataset (NIH, 2003) (see Section 6.1). Another example can be situations where a user is willing to "sell" his/her privacy. For this model, we provide an algorithm that takes as input a random sample from the test set (or the test distribution) and outputs $\widehat{\mathbf{w}}$ that is differentially private with respect to the training data. We also show that $\widehat{\mathbf{w}}$ incurs small prediction error ($O(1/\sqrt{n})$) on the test set, in addition to the error incurred by $\mathbf{w}^*$. (Here n is the size of the training data.) We stress that the "goodness" of $\widehat{\mathbf{w}}$ will be restricted to the test set (or its distribution) only and might not extend to all the possible input points. We demonstrate the practical validity of our theoretical bounds by performing experiments with the polynomial kernel.

Our third model is a non-interactive model where the learner is oblivious to the test data but still sends a $\widehat{\mathbf{w}}$ that is private and guarantees accurate predictions for all possible test points which are in the feature space representation of the domain points. While this model is similar to the model of Chaudhuri et al. (2011), there is a subtle distinction. In the traditional model, we require the difference in predictions $|\langle \mathbf{w}^* - \widehat{\mathbf{w}}, \mathbf{v}\rangle|$ to be small for all $\mathbf{v}$ while in our model we require $|\langle \mathbf{w}^* - \widehat{\mathbf{w}}, \mathbf{v}\rangle|$ to be small only for $\mathbf{v}$'s that are feature space representation of domain points. The kERM approach of Rubinstein et al. (2009); Chaudhuri et al. (2011) addresses this problem to some extent, but their methods are restricted to translation invariant kernels and essentially reduces the kERM to a linear ERM.

Our algorithm for the non-interactive model can also be applied to standard linear ERMs. Here, our sample complexity is $O(d^{1/3})$ compared to $O(d)$ samples required by (Chaudhuri et al., 2011; Rubinstein et al., 2009).

Finally, we provide empirical validation of our semi-interactive method on benchmark datasets using polynomial kernels (where existing methods do not apply directly). Our results show that for reasonably small $\epsilon \approx 0.1$, our method achieves accuracy similar to the non-private baseline classifier. Additionally, in the settings where the dimensionality of the problem is large, and where algorithms of (Chaudhuri et al., 2011) and ours are applicable, experimentally our algorithm we outperform them.

1.1. Contributions:

We consider the problem of differentially private kernelized learning and study it under three practical models. Our algorithms for the first two models are *computationally efficient* but for the third model they can have *exponential* time complexity for some kernel functions.

Interactive: Our interactive model is useful for several learning tasks faced by online systems like ad-systems, recommendation systems. We provide an *efficient* algorithm that can accurately predict for exponentially many test points, in terms of error bound and training points.

Semi-interactive: Our semi-interactive model is useful when public test sets are available. Here, we provide an *efficient* differentially private algorithm with additional generalization error that is independent of the dimensionality of the data.

Non-interactive: Finally, we provide a privacy preserving algorithm with generalization error bound for the standard learning model but where kernel function is restricted to a function of low-dimensional vector spaces. Although our algorithm for this setting might not be computationally efficient in general, but for the case of linear kernels we can prove it to be efficient.

2. Related work

Recently, there has been a lineage of results which try to understand the privacy implications of learning algorithms (Rubinstein et al., 2009; Chaudhuri et al., 2011; Pathak et al., 2010; Williams & McSherry, 2010; Chaudhuri & Hsu, 2011; Kifer et al., 2012; Jain et al., 2012). Typically, the generalization error of these algorithms depends polynomially on the dimensionality of the parameter vector. One notable exception is the method by (Kifer et al., 2012), that achieves logarithmic dependence on the dimensionality when the underlying dataset satisfies Restricted Strong Convexity (RSC) property. Another notable exception is the work of (Chaudhuri et al., 2011), which shows that if the underlying kernel function is translation invariant, then their method outputs a differentially private parameter vector $\widehat{\mathbf{w}}$ with good generalization error. In contrast, our algorithms for releasing private parameter vectors applies to general RKHS where one might not have a translation invariant reproducing kernel.

Another relevant work is by Hall et al. (2012) that provides a method to release function values in RKHS in a differentially private manner. Their techniques can be directly applied to our models but a direct adaptation of their technique suggests significantly worse utility guarantees. For example, using their technique with our interactive model, one can make accurate differentially private predictions for only *quadratically* many (in the number of training points) test points while our method can predict for *exponentially* many test points. Blum et al. (2008) also proposed differentially private techniques for estimating model parameters (or the optima to the ERM). However, their method assumes that the class of possible parameter vectors has a finite number of candidate parameter vectors. Additionally, the running time of the algorithm by Blum et al. (2008) is exponential in the size of the output.

Finally, Gupta et al. (2011); Hardt & Rothblum (2010); Hardt et al. (2010) used online learning techniques to interactively answer linear queries where the error in each query only depends logarithmically on the number of queries. Our algorithms adapt techniques by these methods, for our kERM problem, to provide accurate predictions/parameter vectors in our interactive and semi-interactive models.

3. Preliminaries

Privacy: In this work, we select the notion of differential privacy (Definition 1) for guaranteeing privacy of each of the input data point. Differential privacy is a well studied privacy notion and has emerged as a well accepted definition of privacy for statistical data analysis (Dwork, 2006; 2010). Intuitively, the definition requires that perturbing an individual data point should not lead to any noticeable change in the distribution over the space of possible outputs of a randomized algorithm $\mathcal{A}$.

Definition 1 (Differential privacy (Dwork et al., 2006b;a)). *An algorithm $\mathcal{A}$ is (ϵ, δ)-differentially private if for any two datasets $\mathcal{G}, \mathcal{G}' \in (\mathcal{X} \times \mathcal{Y})^n$ s.t. $\mathcal{G}$ and $\mathcal{G}'$ differ in exactly one data point, and for all measurable sets $\mathcal{O} \subseteq Range(\mathcal{A})$, the following holds:*

$$\Pr[\mathcal{A}(\mathcal{G}) \in \mathcal{O}] \leq e^{\epsilon} \Pr[\mathcal{A}(\mathcal{G}') \in \mathcal{O}] + \delta.$$

Here $(\mathcal{X} \times \mathcal{Y})$ is the domain of the data entries.

Convex Optimization and Kernel Methods: We assume that the loss functions are convex and are of the form: $\ell : \mathbb{R}^{d_\phi} \times (\mathcal{X} \times \mathcal{Y}) \to \mathbb{R}$, where $\mathcal{X}$ is the input domain and $\mathcal{Y}$ is the target output domain. For a given prediction function $f : \mathcal{X} \times \mathbb{R}^{d_\phi} \to \mathbb{R}$, the loss function is alternatively represented as $\ell(f(\mathbf{w}, \mathbf{x}), y)$, where $\mathbf{x} \in \mathcal{X}$, $y \in \mathcal{Y}$ and $\mathbf{w} \in \mathbb{R}^{d_\phi}$. The dimensionality d_ϕ refers to the output dimensionality of the feature map $\phi : \mathcal{X} \to \mathbb{R}^{d_\phi}$. For the chosen feature map ϕ, we assume that there exists a kernel function K which can efficiently compute the inner product of any $\phi(\mathbf{x})$ and $\phi(\mathbf{v})$ as $K(\mathbf{x}, \mathbf{v}) = \langle \phi(\mathbf{x}), \phi(\mathbf{v}) \rangle$, where $\mathbf{x}, \mathbf{v} \in \mathcal{X}$. Additionally, we assume that the feature map ϕ is L_ϕ-Lipschitz and the loss function ℓ is L-Lipschitz in the first parameter. We denote vectors in bold ($\mathbf{w}$) while matrices with capital letters (X).

4. Problem Formulation

In this section, we study the problem of differentially private learning using regularized empirical risk minimization (ERM) in kernel space. ERM is a canonical learning method for several concept classes; regularization helps in avoiding over-fitting to the training data and guaranteeing good generalization error. In general, a regularized ERM can be written as the following optimization problem:

$$\min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} \ell(f(\mathbf{w}, \mathbf{x}_i); y_i)) + \frac{\lambda}{2} r(\mathbf{w}), \tag{1}$$

where $f : \mathcal{X} \times \mathbb{R}^{d_\phi} \to \mathbb{R}$ is the prediction function, $\mathbf{w} \in \mathbb{R}^{d_\phi}$ are the prediction function parameters, and $(\mathbf{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ are training samples from a fixed (unknown) distribution $\mathcal{P}$, i.e., $(\mathbf{x}_i, y_i) \sim \mathcal{P}$. $\mathcal{X}$ is the input domain, while $\mathcal{Y}$ is the prediction domain. For example, $\mathcal{Y} = \{-1, +1\}$ for classification problem. Here, ℓ is the (surrogate) convex loss function for the predictions $f(\cdot, \cdot)$.

Typically, f is selected to be a linear prediction function, i.e., $f(\mathbf{w}, \mathbf{x}_i) = \langle \mathbf{w}, \mathbf{x}_i \rangle$. However, several practical applications have complex decision boundaries that cannot be captured by linear predictors. In fact for several applications (e.g., computer vision) obtaining a vector representation for data points itself might not be possible. A common approach to circumvent this problem is by using "kernel trick". That is each $\mathbf{x}_i$ is mapped to a higher dimensional *feature space* and f

is required to be linear in that mapped feature space. That is, $f(\mathbf{w}, \mathbf{x}_i) = \langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle$, where $\phi : \mathcal{X} \to \mathbb{R}^{d_\phi}$ and d_ϕ is the dimensionality of the mapped feature space. Also, $K(\mathbf{x}, \mathbf{v}) = \langle \phi(\mathbf{x}), \phi(\mathbf{v}) \rangle$ is assumed to be an efficiently computable kernel function.

In this work, we restrict our ERM formulation to L_2 regularization, i.e., $r(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|_2^2$. Formally, we consider the following general ERM:

$$\mathbf{w}^* = \arg \min_{\mathbf{w} \in \mathbb{R}^{d_\phi}} \frac{1}{n} \sum_{i=1}^{n} \ell(\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle ; y_i) + \frac{\lambda}{2} \|\mathbf{w}\|_2^2, \quad (2)$$

where d_ϕ is the dimensionality of mapped feature space, $\lambda > 0$ is the regularization parameter. Note that $\mathbf{w}^*$ is dependent on each of the data point and observing it might lead to privacy compromise of an individual point $(\mathbf{x}_i, y_i)$. Hence, the goal is to provide predictions that are similar to predictions using optimal $\mathbf{w}^*$ while preserving privacy of each individual point. To guarantee privacy, we use the well-known notion of differential privacy (Definition 1).

Recently Chaudhuri et al. (2011); Rubinstein et al. (2009) proposed algorithms for approximately solving (2) while preserving differential privacy. However, their algorithms are either restricted to linear decision functions ($f(\mathbf{w}, \mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle$) (which has polynomial dependence of the error on the dimensionality of the feature space) or non-linear decision functions using a restricted class of translation-invariant kernels, e.g., the Gaussian kernel.

In contrast, we address the above mentioned problem of privately learning from data points in arbitrary RKHS. Now, $\mathbf{w}^*$ is a d_ϕ dimensional vector, where d_ϕ can potentially be very large and might not even be representable explicitly. Kernel methods exploit the fact that $\mathbf{w}^*$ is of the form $\mathbf{w}^* = \sum_i \mu_i \phi(\mathbf{x}_i)$ and hence maintain $\mu_i, 1 \leq i \leq n$ to store/use $\mathbf{w}^*$. As observed by Chaudhuri et al. (2011), releasing $\mathbf{w}^*$ in this case is hard, since the learner does not have explicit access to $\mathbf{w}^*$ and in fact relies on each training point (or support vector) $\mathbf{x}_i$ to compute prediction values. Hence, the "traditional" model of releasing a differentially private $\widehat{\mathbf{w}}$ which approximates $\mathbf{w}^*$ over the entire domain $\mathcal{X}$ might be too restrictive for this problem.

In the next section, we consider an interactive model that does not release $\mathbf{w}^*$ explicitly but provides predictions similar to $\mathbf{w}^*$ while preserving privacy. Then in Section 6.1, we provide a semi-interactive model where a differential private version of $\mathbf{w}^*$ is released but accuracy in its predictions are guaranteed only for the data coming from a fixed distribution. Finally in Section 6.2, we provide a method for our non-interactive model where the learner releases a differentially private version of $\mathbf{w}^*$ with guaranteed accurate predictions (compared to $\mathbf{w}^*$) over entire input domain $\mathcal{X}$. However, here, we need to restrict input space $\mathcal{X}$ and also *unlike* our other methods, this method might require exponential (in n) amount of computation time. See Figure 1 for a block schematic of our three models.

5. Interactive Model

In this section, we describe our interactive model for releasing privacy preserving predictions for given test points. Our model consists of three parties: a dataset, a learner, and a user. A trusted *learner* obtains supervised data from the *dataset* and learns model parameters $\mathbf{w}^*$ by solving (2). Then, *user* sends its test points to the learner (in online/batch mode) and the learner provides predictions that preserve privacy of each training point and are also close to the predictions obtained using $\mathbf{w}^*$.

Example Scenario 1: Consider the case of an sponsored search ad system where the ad delivery engine needs to find the relevance of a particular ad for a given user query. Now, typically ad engines use stored user-click logs to learn a classifier for such tasks. In the context of our differentially private interactive model, the goal of the ad engine (learner) would be to predict relevance of a given ad, user query pair (user test point) accurately while preserving privacy of the training data, i.e, user-click logs (dataset).

Similar privacy preserving learning scenarios can be found in several other online systems as well, such as recommendation system, social networks etc.

Example Scenario 2: Consider two hospitals A and B. Hospital A sends its *labeled* data (health profile of its patients) to a trusted research lab (learner) that learns a classifier using the supplied data by A. Then, Hospital B sends its *unlabeled* data to the trusted lab (learner), which returns back reasonably accurate labels for the test data from B while guaranteeing privacy to each data point from A.

We now formalize our model. The dataset provides labeled samples $\mathcal{G} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ to the learner using which learner estimates $\mathbf{w}^*$ by solving (2). The user provides its test points over T rounds. During each round, the user provides the learner a test point $\mathbf{z}_t \in \mathcal{X}$ for which the learner predicts $\widehat{y}_t$ that is differentially private w.r.t. $\mathcal{G}$. The learner should ensure that with high probability, $\forall t$, $|\widehat{y}_t - \langle \mathbf{w}^*, \phi(\mathbf{z}_t) \rangle| \leq \sigma$, where σ is a fixed parameter.

For this problem, we adopt a recent technique from the literature of private interactive dataset release: *iterative dataset construction* (IDC) method (Gupta et al., 2011). Gupta et al. (2011) show that their IDC based method can be used to release accurate answers to linear queries over a private dataset. In the context of our problem, the private "dataset" is $\mathbf{w}^*$ and the goal is to answer linear queries $\widehat{y}_t = f(\mathbf{w}; \mathbf{z}_t) = \langle \mathbf{w}, \mathbf{z}_t \rangle$. At

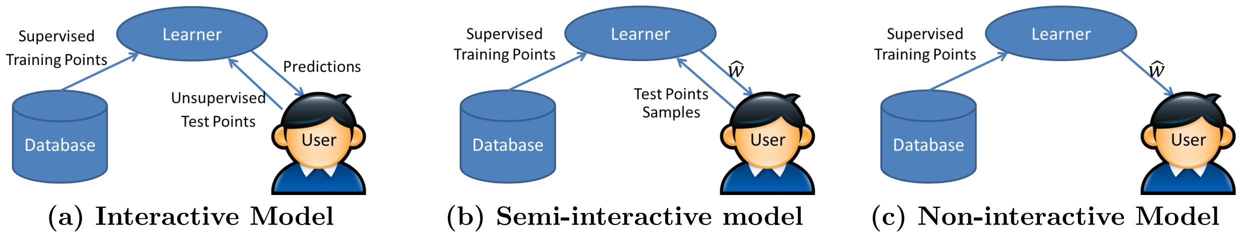


Figure 1. Models for kernelized privacy preserving learning using kernel ERM. We have three parties: a dataset, a trusted learner and a user. *Learner* learns optimum ($\mathbf{w}^*$) of the ERM using the training data from the *dataset*. User's goal is to obtain labels for its test set while learner's goal is to provide user with accurate predictions/model parameters without violating dataset's privacy. **(a) Interactive Model**: In this model, the user sends its test data to the learner for which it returns back accurate predictions without violating dataset's privacy. **(b) Semi-interactive model**: In this model, the user sends a small subset of its test set, and then learner sends a differentially private $\widehat{\mathbf{w}}$ that is guaranteed to obtain similar predictions to $\mathbf{w}^*$ on user's test set. **(c) Non-interactive Model**: In this model, learner sends the user a differentially private $\widehat{\mathbf{w}}$ that is expected to provide similar predictions to $\mathbf{w}^*$ on all the points in the input space.

a high level, the algorithm at each step maintains a differentially private version $\mathbf{w}_t$ of the true dataset $\mathbf{w}^*$. For a given query $\mathbf{z}_t$, algorithm first tries to answer using $\mathbf{w}_t$, i.e., $\widehat{y}_t = \langle \mathbf{w}_t, \mathbf{z}_t \rangle$. However, if the prediction is inaccurate w.r.t. $\mathbf{w}^*$, i.e., $|\langle \mathbf{w}_t, \mathbf{z}_t \rangle - \langle \mathbf{w}^*, \mathbf{z}_t \rangle| > \sigma$, then $\widehat{y}_t$ is predicted using true $\mathbf{w}^*$ with appropriately added noise, i.e., $\widehat{y}_t = \langle \mathbf{w}_t, \mathbf{z}_t \rangle + \zeta_t$ for appropriately calibrated noise ζ_t. Also, in this case, the algorithm updates $\mathbf{w}_t$ so that it gets "closer" to $\mathbf{w}^*$.

Main Algorithmic Idea: In the case of learning with kernels, we cannot explicitly maintain $\mathbf{w}_t$. However, it is easy to see that $\mathbf{w}_t$ obtained using IDC updates is of the form $\mathbf{w}_t = \sum_{\tau=1}^{t} \alpha_\tau \phi(\mathbf{z}_\tau)$. Further, each $\mathbf{z}_t$ is public (to the user). Hence, to maintain differentially private $\mathbf{w}_t$, we need to maintain differentially private α_t's only. Algorithm 1 (Algorithm PINP) provides a pseudo-code of our method for releasing predictions in the case of (non-linear) kernels. Below, we provide both privacy as well as utility (i.e., accuracy in prediction w.r.t. $\mathbf{w}^*$) guarantees for PINP.

Theorem 2 (Privacy Guarantee)**.** *Let $\mathbf{w}^*$ be the optimal solution to (2) and let $\widehat{y}_t$ be predicted by Algorithm 1 for a given point $\mathbf{z}_t$. Then, each output $\widehat{y}_t, 1 \le t \le T$ and hence Algorithm 1 is (ϵ, δ)-differentially private w.r.t. input training samples $\mathcal{G} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$.*

See supplementary material for a proof of Theorem 2.

Utility analysis of Algorithm 1

Theorem 3 (Error bound)**.** *Let $\widehat{y}_t, 1 \le t \le T$ be the prediction by Algorithm 1 for the t-th step test point $\mathbf{z}_t$. Let $\mathbf{w}^* = \sum_i^n \mu_i \phi(\mathbf{x}_i)$ be the optimal solution to (2) with $\|\mathbf{w}^*\|_2 \le C$. Then, with probability at least $1-\beta$, for each prediction $\widehat{y}_t$, the error incurred by Algorithm 1 is bounded by:*

$$|\widehat{y}_t - \langle \phi(\mathbf{z}_t), \mathbf{w}^* \rangle| = O\left(\frac{CR_\phi^2 \sqrt{L \log(T/\beta)} \log^2 \frac{1}{\delta}}{\sqrt{n\epsilon\lambda}} \right).$$

The above theorem shows that the error in Algorithm 1 depends only logarithmically on the number of test points (T), while increasing the number of training points (n) rapidly decreases error incurred. See Appendix A.2 for a proof sketch of the above theorem.

Note that we assume a differentially private bound C on $\|\mathbf{w}^*\|_2$. A simple bound of $C = \frac{2LR_\phi}{\lambda}$ follows directly using optimality of $\mathbf{w}^*$. Otherwise, C can also be estimated differentially privately by adding a small noise to $\|\mathbf{w}^*\|_2$. For clarity, in the later sections we assume that $\|\mathbf{w}^*\|_2$ is *available publicly* and hence can be used by our algorithms without violating privacy.

6. Non-Interactive Setting

In the previous section we presented an algorithm that provides prediction $(f(w^*; \mathbf{z}) = \langle w^*, \phi(\mathbf{z}) \rangle)$ for each of the test samples $\mathcal{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_T\}$ within $O(\sqrt{(\log T)/n})$ error while guaranteeing privacy for each of the n training samples ($\mathcal{G}$). However, in this model the "learner" itself needs to answer each test query which might be undesirable for both "learner" and the "user".

In some applications, it would be useful for the "learner" to release a differentially private version of $\mathbf{w}^*$ so that it would have similar generalization error as $\mathbf{w}^*$ on test samples of the "user". As mentioned earlier in Section 4, the learner maintains $\mathbf{w}^*$ only implicitly in terms of training data points $(\mathbf{x}_i, y_i)$. Hence, it seems impossible for the learner to privately release $\mathbf{w}^*$ unless it makes assumption about the test samples or makes assumptions about the input space $\mathcal{X}$ and/or the kernel space K.

(Chaudhuri et al., 2011) took the later approach for differentially private kernel ERM, i.e., they assumed that $\mathcal{X}$ is a finite dimensional vector space and K is translation invariant. However, for several applications one or both of the above mentioned constraints may not be satisfied.

In this section we propose two different models for

Algorithm 1 Private Interactive Non-linear Prediction Algorithm (PINP)

```
Require: Optimum of (2): $\mathbf{w}^* = \sum_{i=1}^n \mu_i \phi(\mathbf{x}_i)$; Test points: $\mathcal{Z} = \{\mathbf{z}_1, \cdots, \mathbf{z}_T\}$, $\forall t,\ \mathbf{z}_t \in \mathcal{X}$; Kernel: $K(\mathbf{x}, \mathbf{v}) = \langle \phi(\mathbf{x}), \phi(\mathbf{v}) \rangle$; $R_\phi = \max_{\mathbf{z} \in \mathcal{X}} \|\phi(\mathbf{z})\|_2$; Privacy parameters: $(\epsilon, \delta)$; failure probability $\beta$; Non-private bound $C$: $\|\mathbf{w}^*\|_2 \le C$; Lipschitz constant of $\ell$: $L$; regularization parameter: $\lambda$.
 1: Set the bound on the number of updates: $B \leftarrow \frac{n\lambda\epsilon C^2}{200 L R_\phi \sqrt{2\log(2T/\beta)}\log(4/\delta)}$
 2: Noise parameter: $\epsilon_0 \leftarrow \frac{n\epsilon\lambda}{200 L R_\phi^2 \sqrt{B}\log(4/\delta)}$, error: $\sigma \leftarrow \frac{4}{\epsilon_0}\log(2T/\beta)$, step size: $\eta \leftarrow \frac{\sigma}{4R_\phi^2}$
 3: Set $\alpha_0 \leftarrow 0$    //Implicitly assume, $\mathbf{w}_0 = 0$
 4: for $t \in \{1, \cdots, T\}$ and $counter < B$ do
 5:   $\widehat{a}_t \leftarrow \langle \mathbf{w}^*, \phi(\mathbf{z}_t) \rangle + \zeta_t = \sum_{i=1}^n \mu_i K(\mathbf{z}_t, \mathbf{x}_i) + \zeta_t$, $\zeta_t \sim Lap(1/\epsilon_0)$
 6:   $\widehat{d}_t \leftarrow \widehat{a}_t - \sum_{\tau=0}^{t-1} \alpha_\tau K(\mathbf{z}_t, \mathbf{z}_\tau)$
 7:   if $|\widehat{d}_t| > \sigma$ then
 8:     $\alpha_t \leftarrow \eta \mathrm{sign}(\widehat{d}_t)$ and $counter \leftarrow counter + 1$
          //Implicitly, $\mathbf{w}_t = \sum_{\tau=1}^t \alpha_\tau \phi(\mathbf{z}_\tau)$
 9:     Output prediction: $\widehat{y}_t = \widehat{a}_t$
10:   else
11:     $\alpha_t \leftarrow 0$
12:     Output prediction: $\widehat{y}_t = \sum_{\tau=1}^t \alpha_\tau K(\mathbf{z}_\tau, \mathbf{z}_t)$
13:   end if
14: end for
```

this problem: 1) Test Data *Dependent* Learner (*Semi-interactive model*) and 2) Test Data *Independent* Learner (*Non-interactive model*). In the first model (see Section 6.1), we assume that the learner is provided with a small set of random samples from the test data and then the learner provides a differentially private $\widehat{\mathbf{w}}$ with provable error bounds over the test set. In this model, we do not make any assumption about the input space $\mathcal{X}$ or the kernel function K. Our second model is agnostic to the test data, i.e., it does not seek any test samples, but it needs to assume $\mathcal{X}$ to be a low-dimensional vector space (i.e., the dimensionality has to be lesser than $O(n^2)$, see Section 6.2 for more details). Furthermore, unlike our algorithm for the semi-interactive setting, our algorithm (for the non-interactive setting) can potentially take exponential amount of time in the dimensionality of $\mathcal{X}$. One notable exception is the case of linear kernels, where we can provide an efficient algorithm for our non-interactive setting as well. Also note that, for linear kernels, non-interactive setting is essentially the standard setting of differentially private ERM introduced by (Chaudhuri et al., 2011).

6.1. Test Data Dependent Learner (Semi-interactive model)

In this section, we propose our *test data dependent* model for differentially private release of the optimal solution $\mathbf{w}^*$ to the regularized kernel based ERM (2). In this model, we assume that the user sends a small subset $\mathcal{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_T\}$ of its test data points $\mathcal{Q} = \{\mathbf{q}_1, \cdots\}$ to the learner and the learner in turn sends $\widehat{\mathbf{w}}$ which is: 1) differentially private w.r.t. training data $\mathcal{G}$ and 2) has small excess error on the test data in addition to the error incurred by $\mathbf{w}^*$. Figure 1 (b) shows our semi-interactive model.

Example scenario: This scenario is motivated by the HapMap project of the National Institute of Health (NIH). HapMap (NIH, 2003) is a *public* dataset that contains genetic data from four populations with African, Asian, and European ancestry. Several genomic research labs, with great effort, collect labeled genetic data using which they can learn a good (non-linear) classifier (for say a particular disease). In the context of our semi-interactive model, the private genetic data of the lab is the private *dataset* and the lab is the *learner*. While the lab (learner) would like to (or is required to) predict disease for new data points (test points), it wouldn't like to violate privacy of its own data. Hence, a privacy preserving mechanism would be required to release a reasonably accurate (non-linear) classifier. While this task for general kernels seems infeasible, one can exploit the publicly available HapMap data to release differentially private classifier that guarantees good accuracy on sample points "similar" to the HapMap dataset. That is, the HapMap dataset would be the sample from test set that our model requires.

Our Solution : The main idea of our solution is to learn a differentially private $\hat{\mathbf{w}}$ that gives "similar" prediction to $\mathbf{w}^*$ on the dataset $\mathcal{Z}$ while also preserving privacy. We then use standard stochastic optimization based generalization error guarantees to argue for excess error incurred by $\hat{\mathbf{w}}$ for the entire test domain. To this end, we first ensure that each prediction over $\mathcal{Z}$ is differentially private by adding appropriate noise b_t for all t. We then learn $\hat{\mathbf{w}}$ by solving a simple least squares problem: $\hat{\mathbf{w}} = \arg\min_{\mathbf{w}\in\mathcal{C}} \frac{1}{T}\sum_{t=1}^T \left(\langle \mathbf{w}, \phi(\mathbf{z}_t)\rangle - \langle \mathbf{w}^*, \phi(\mathbf{z}_t)\rangle - b_t\right)^2$. Note that b_t typically has to scale as $O(\sqrt{T})$ in the above mentioned approach; one can use algorithm given in the previous section to add only $O(\log T)$ noise per b_t. However, the analysis is more involved and it doesn't provide significantly stronger utility guarantees.

Below, we show that $\widehat{\mathbf{w}}$ returned by our algorithm preserves privacy of training data $\mathcal{G}$ as well as incurs small additional error on Q compared to $\mathbf{w}^*$.

Privacy: The privacy guarantee follows directly from the composition property of differential privacy (Dwork et al., 2010); see supplementary material.

Theorem 4. *Algorithm 2 is (ϵ, δ)-differentially pri-*

Algorithm 2 Test Data Dependent Learner (TDDL)

```
Require: Optimal solution to (2): $\mathbf{w}^* = \sum_{i=1}^{n} \mu_i \phi(\mathbf{x}_i)$; $K(\mathbf{x}, \mathbf{v}) = \langle \phi(\mathbf{x}), \phi(\mathbf{v}) \rangle$; $R_\phi = \max_{\mathbf{z} \in \mathcal{X}} \|\phi(\mathbf{z})\|_2$; Privacy parameters: $(\epsilon, \delta)$; Lipschitz constant of $\ell$: $L$; regularization parameter: $\lambda$; $\mathcal{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_T\}$ uniformly sampled from test set $\mathcal{Q}$; convex set $\mathcal{C}$.
1: Sample random entries $b_1, \cdots, b_T$ i.i.d. from Lap($\nu$), where $\nu = O\left(\frac{L{R_\phi}^2 \sqrt{T \log(1/\delta)}}{\lambda n \epsilon}\right)$.
2: Output $\hat{\mathbf{w}} = \arg\min_{\mathbf{w} \in \mathcal{C}} \frac{1}{T} \sum_{t=1}^{T} \left( \langle \mathbf{w} - \mathbf{w}^*, \phi(\mathbf{z}_t) \rangle - b_t \right)^2$.
```

vate.

Utility: The theorem below shows that given a *fixed* set Q, the learner can supply a differentially private $\widehat{\mathbf{w}}$ that incurs small loss in additional to the loss incurred by $\mathbf{w}^*$. See supplementary material for a proof.

Theorem 5 (Error bound)**.** *Let $\mathcal{Q} = \{\mathbf{q}_1, \dots, \mathbf{q}_{|Q|}\}$ be the test set, let $\mathcal{Z} = \{\mathbf{z}_1, \cdots, \mathbf{z}_T\}$ be sampled uniformly at random from $\mathcal{Q}$. If $T = O\left((\|\mathcal{C}\|_2 n\epsilon\lambda)/(LR_\phi\sqrt{\log(1/\delta)})\right)$ and $\mathbf{w}^* \in \mathcal{C}$ in Algorithm 2, then w.p. $1-\beta$,*

$$\frac{1}{|Q|}\sum_{i=1}^{|Q|} \ell(\langle \widehat{\mathbf{w}}, \phi(\mathbf{q}_i)\rangle; y_{\mathbf{q}_i}) \leq \frac{1}{|Q|}\sum_{i=1}^{|Q|} \ell(\langle \mathbf{w}^*, \phi(\mathbf{q}_i)\rangle; y_{\mathbf{q}_i}) + O\left(\frac{(LR_\phi)^{3/2}\sqrt{\|\mathcal{C}\|_2 \log^{1/2}(1/\delta)}\log(T/\beta)}{\sqrt{n\epsilon\lambda}}\right).$$

Here $y_{\mathbf{q}_i}$ refers to the label for the test point $\mathbf{q}_i$.

We can also easily generalize our result from finite set Q to the general distribution $\mathcal{P}$ from which $\mathcal{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_T\}$ is sampled. See Theorem 13 (Appendix B.3) for an exact statement.

Next, we provide a generalization error bound for our privacy preserving algorithm when both training and test points are sampled from the same distribution.

Corollary 6 (Generalization Error Bound)**.** *Let both training and test samples be i.i.d. samples from $\mathcal{P}$. Let $\mathcal{C} = \{\mathbf{w} : \|\mathbf{w}\|_2 \leq 2LR_\phi/\lambda\}$, where R_ϕ and λ are as defined in Algorithm 2, $T = O\left((n\epsilon)/(\sqrt{\log(1/\delta)})\right)$, and let $n = \Omega^*\left(\frac{L^4 R_\phi^3 \log^2 \frac{1}{\beta} \log^{1/2} \frac{1}{\delta}}{\epsilon_g^2 \epsilon}\right)$. Then, for $\widehat{\mathbf{w}}$ obtained using Algorithm 2, we have (w.p. $1-\beta$):* $\mathbb{E}_{\mathbf{z}\sim\mathcal{P}}[\ell(\langle \widehat{\mathbf{w}}, \phi(q_i)\rangle; y_{q_i})] \leq \epsilon_g$.

Note that the above sample complexity of our Algorithm 2 is *independent* of the dimensionality. In contrast, existing privacy preserving learning methods like (Chaudhuri et al., 2011) has polynomial dependence on the dimensionality. Naturally, key difference is that we have an additional requirement of a small test data set sample. Another way to understand this difference is that our method guarantees that $|\ell(\widehat{\mathbf{w}}; \mathbf{z}) - \ell(\mathbf{w}^*; \mathbf{z})|$ is small *only* if $\mathbf{z}$ is sampled from $\mathcal{P}$. However, these algorithms can guarantee that $|\ell(\widehat{\mathbf{w}}; \mathbf{z}) - \ell(\mathbf{w}^*; \mathbf{z})|$ is small for *all* $\mathbf{z} \in \mathbb{R}^d$.

6.2. Test Data Independent Learner (Non-interactive model)

In the previous section, we proposed a model where the *user* needs to send a random sample of its test data to obtain a differentially private $\widehat{\mathbf{w}}$ that is expected to perform well on all of the test set. However, this model is not applicable in scenarios where the user cannot or does not want to share his test data with a trusted third party, i.e., with the *learner*.

In this section, we analyze the later setting where the user does not have access to unlabeled samples. Figure 1 (c) shows our model. The model is same as traditional differentially private ERM model (Chaudhuri et al., 2011), where a learner sends a differentially private version of the optimum $\mathbf{w}^*$ (*i.e.*, $\widehat{\mathbf{w}}$) to the user.

For this model, our approach is similar to that in Section 6.1 except that the algorithm itself *generates* a sequence of test samples $\mathcal{Z} = \{\mathbf{z}_1, \cdots, \mathbf{z}_T\}$. The goal is that the generated samples $\mathbf{z}_t'$s should be able to *distinguish* the current iterate $\mathbf{w}_t = \sum_{i=1}^{t} \alpha_i \phi(\mathbf{z}_i)$ from the underlying parameter vector $\mathbf{w}^*$. That is, $\mathbf{z}_t$'s force Algorithm 1 to make updates to its differentially private model parameters $\mathbf{w}_t$. Furthermore, we require to find $\mathbf{z}_t$'s, in a differentially private manner. To this end, we use the exponential mechanism, a well-studied technique in the privacy literature (McSherry & Talwar, 2007). Broadly speaking, we sample $\mathbf{z}_t$ from a distribution s.t. $P(\mathbf{z}) \propto \exp\left(\frac{\epsilon_0 n\lambda}{8LR_\phi^2} |\langle \phi(\mathbf{z}), \mathbf{w}_t - \mathbf{w}^*\rangle|\right)$. That is, probability of sampling a $\mathbf{z}$ is higher if $|\langle \phi(\mathbf{z}), \mathbf{w}_t - \mathbf{w}^*\rangle|$ is large, i.e., if $\mathbf{z}$ can "distinguish" $\mathbf{w}_t$ and $\mathbf{w}^*$. Also, to ensure differential privacy, the distribution introduces appropriate randomness. See Algorithm 3 for a pseudo-code of our approach. Note that, we assume that the input space is restricted to be a vector space, i.e., each $\mathbf{z}_t \in \mathbb{R}^d$, $\|\mathbf{z}_t\|_2 \leq 1$.

Now, we provide the privacy and utility guarantees.

Theorem 7. *Algorithm 3 is (ϵ, δ)-differentially private.*

Broadly, our proof follows by combining the analysis of exponential mechanism (McSherry & Talwar, 2007) and the privacy proof of Algorithm 1 (Theorem 2). See supplementary material for a proof sketch. Next we show that the predictions obtained using $\widehat{\mathbf{w}}$ given by Algorithm 3 is "similar" to the ones obtained using $\mathbf{w}^*$ for *all* points $\mathbf{z} \in \mathcal{X}$, where $\mathcal{X} = \{\mathbf{z} \in \mathbb{R}^d \ s.t. \ \|\mathbf{z}\|_2 \leq 1\}$ is the input space.

Algorithm 3 Test Data-independent Learner (TDIL)

```
Require: Optimal solution to (2): $\mathbf{w}^* = \sum_{i=1}^{n} \mu_i \phi(\mathbf{x}_i)$, Privacy parameters: $(\epsilon, \delta)$;
    $R_\phi = \max_{\mathbf{z} \in Z} \|\phi(\mathbf{z})\|_2$; $\beta$: failure probability, $L$: Lipschitz bound on the loss function $\ell$; $\lambda$:
    regularization parameter (in (2)).
 1: Bound on number of updates: $B \leftarrow \frac{n\epsilon}{100 L R_\phi \log(4/\delta)}$,
    $\epsilon_0 \leftarrow \frac{n\epsilon\lambda}{10 L R_\phi^2 \sqrt{B} \log(4/\delta)}$, $\sigma \leftarrow \frac{4}{\epsilon_0}\log(2B/\beta)$,
    step size: $\eta \leftarrow \frac{\sigma}{4R_\phi^2}$, net size: $\nu = \frac{dR_\phi}{\epsilon_0 n L_\phi}$
 2: Set $\alpha_0 \leftarrow 0, \cdots, \alpha_B \leftarrow 0$, $\mathbf{w}_0 \leftarrow 0$
 3: Let $S$ be $\nu$-net on the ball $\{\mathbf{z} : \|\mathbf{z}\|_2 \le r_\phi\}$.
 4: for $t = \{1, \cdots, B\}$ do
 5:   Sample $\mathbf{z} \in S$ w.p. $\exp(\frac{\epsilon_0 n\lambda}{8LR_\phi^2} | \langle \phi(\mathbf{z}), \mathbf{w}_t - \mathbf{w}^* \rangle |)$
 6:   $\widehat{d}_t = \langle \mathbf{w}^*, \phi(\mathbf{z}) \rangle - \sum_{i=0}^{t} \alpha_t \phi(\mathbf{z})^T \phi(\mathbf{z}_t) + Lap(\frac{1}{2\epsilon_0})$
 7:   if $|\widehat{d}_t| > \sigma$ then
 8:     $\alpha_t \leftarrow \eta \mathrm{sign}(\widehat{d}_t)$. Output: $\widehat{a}_t \leftarrow \langle \mathbf{w}^*, \phi(\mathbf{z}_t) \rangle + Lap(\frac{1}{2\epsilon_0})$.
 9:   else
10:     $\alpha_i \leftarrow 0$. Output : $\sum_{i=0}^{t} \alpha_i \phi(\mathbf{z}_i)^T \phi(\mathbf{z}_t)$.
11:   end if
12:   $\mathbf{z}_t \leftarrow \mathbf{z}$ and $\mathbf{w}_{t+1} \leftarrow \sum_{i=1}^{t} \alpha_i \phi(\mathbf{z}_i)$.
13: end for
14: Output: $\widehat{w} = \mathbf{w}_B$.
```

Theorem 8 (Error bound)**.** *Let L_ϕ be the Lipschitz constant for the feature map ϕ, $\mathcal{X} = \{\mathbf{z} \in \mathbb{R}^d$ s.t. $\|\mathbf{z}\|_2 \le 1\}$ and $\mathbf{w}^*$ be the optimal solution to (2). Then, with probability at least $1 - \beta$, the output of Algorithm 3 ($\widehat{\mathbf{w}}$) satisfies:*

$$\ell(\langle \phi(z), \widehat{\mathbf{w}} \rangle ; y_z) \le \ell(\langle \phi(z), \mathbf{w}^* \rangle + \frac{C_1 R_\phi d^{1/3} L^{8/3} \ln L_\phi \log \frac{1}{\beta}}{(\lambda^2 n)^{2/3} \sqrt{\epsilon} (1/\log^2 \frac{1}{\delta})},$$

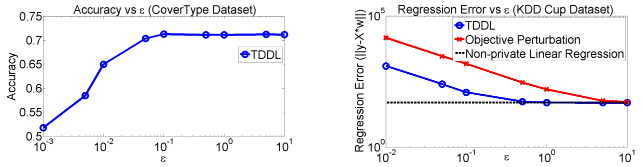
for all $\mathbf{z} \in \mathbb{R}^d$ and $\|\mathbf{z}\|_2 \le 1$. $C_1 > 0$ is a global constant.

See supplementary material for a proof. We note that although the dependence of the generalization error is worse with respect to the size of the training set (i.e., $n^{2/3}$ as compared to n in (Chaudhuri et al., 2011)), the error in our case depends only on the dimensionality (d) of the low-dimensional space ($\mathcal{X}$) instead of the dimensionality (d_ϕ) of the kernel space as in (Chaudhuri et al., 2011).

We would like to stress that Algorithm 3 may not be always computationally efficient. However, for the special case of linear kernels, an efficient version can be constructed via sampling from a uniform mixture of two log-concave distributions.

7. Experiments

In this section, we present experimental evaluation of our TDDL method for the semi-interactive model (Algorithm 2). The goal of these experiments is to demonstrate that our TDDL method, while guaranteeing privacy, is practical and do not deteriorate accuracy significantly for reasonable privacy requirements. Additionally, we demonstrate that our algorithm is significantly more accurate than (Chaudhuri et al., 2011) for high-dimensional datasets in the traditional linear kernel setting.



(a) (b)

Figure 2. Results for our TDDL algorithm (Algorithm 2). a): Classification accuracy achieved by TDDL (with polynomial kernel) as the privacy parameter ϵ varies. Baseline accuracy of the non-private learner is 72.3% b): Regression error achieved by different methods on the KDD Cup 2010 (algebra) dataset. Clearly, our TDDL method significantly outperforms the Objective Perturbation method by (Chaudhuri et al., 2011) and is able to get accuracies close to baseline ($\epsilon \ge 0.5$.)

For our first experiment, we applied our method to the CoverType dataset (see supplementary material for results on the URL Reputation dataset). We used nonlinear SVM as our classifier and trained them using $500K$ training examples. We selected a third-degree polynomial kernel as our kernel function, $K(\mathbf{x}, \mathbf{y}) = (\langle \mathbf{x}, \mathbf{y} \rangle + 1)^3$. Note that, this kernel is not translation invariant and hence the method of (Chaudhuri et al., 2011) does not apply for this problem. We used LibSVM (Chang & Lin, 2011) for training SVMs and report results averaged over 5 runs. The penalty parameter for SVM training was set to be $C = 0.001$.

Figure 2 (a) plots our methods results for the CoverType dataset. Baseline accuracy of the non-private learner is 72.3%. Note that for $\epsilon \ge 0.1$, our method performed similarly to the baseline method.

Next, we study our algorithms in the linear but *high-dimensional* model. For this, we selected the KDD Cup 2010 dataset that contains data points embedded in around $2M$ dimensions. We selected 20000 training points randomly and learned a linear regression function. Figure 2 (b) compares ℓ_2 regression error ($\|X\mathbf{w} - y\|_2$) incurred by our method with the Objective Perturbation method of (Chaudhuri et al., 2011) and the non-private least squares method. Clearly, our method achieves significantly smaller error when compared to the method of (Chaudhuri et al., 2011). The reason is that our method adds noise whose magnitude is independent of the dimensionality d, while for (Chaudhuri et al., 2011) the norm of the noise varies linearly with d. We also stress that our method assumes semi-interactive model where it has access to a small subset of the test set; in contrast, (Chaudhuri et al., 2011) doesn't exploit such a subset and hence as such is at a disadvanatage.

References

Blum, Avrim, Ligett, Katrina, and Roth, Aaron. A learning theory approach to non-interactive database privacy. In *STOC*, pp. 609–618. ACM, 2008.

Chang, Chih-Chung and Lin, Chih-Jen. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2: 27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Chaudhuri, Kamalika and Hsu, Daniel. Sample complexity bounds for differentially private learning. *Journal of Machine Learning Research - Proceedings Track*, pp. 155–186, 2011.

Chaudhuri, Kamalika, Monteleoni, Claire, and Sarwate, Anand D. Differentially private empirical risk minimization. *JMLR*, 12:1069–1109, 2011.

Dwork, Cynthia. Differential privacy. In *ICALP (2)*, 2006.

Dwork, Cynthia. Differential privacy in new settings. In *SODA*, 2010.

Dwork, Cynthia, Kenthapadi, Krishnaram, Mcsherry, Frank, Mironov, Ilya, and Naor, Moni. Our data, ourselves: Privacy via distributed noise generation. In *In EUROCRYPT*, pp. 486–503. Springer, 2006a.

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

Dwork, Cynthia, Rothblum, Guy N, and Vadhan, Salil. Boosting and differential privacy. In *FOCS*, 2010.

Grauman, Kristen and Darrell, Trevor. The pyramid match kernel: Efficient learning with sets of features. *Journal of Machine Learning Research*, 8:725–760, 2007.

Gupta, Anupam, Roth, Aaron, and Ullman, Jonathan. Iterative constructions and private data release. *CoRR*, abs/1107.3731, 2011.

Hall, Rob, Rinaldo, Alessandro, and Wasserman, Larry A. Differential privacy for functions and functional data. *CoRR*, abs/1203.2570, 2012.

Hardt, Moritz and Rothblum, Guy N. A multiplicative weights mechanism for privacy-preserving data analysis. In *FOCS*, 2010.

Hardt, Moritz, Ligett, Katrina, and McSherry, Frank. A simple and practical algorithm for differentially private data release. *CoRR*, abs/1012.4763, 2010.

Jain, Prateek, Kothari, Pravesh, and Thakurta, Abhradeep. Differentially private online learning. In *COLT*, 2012.

Kifer, Daniel, Smith, Adam, and Thakurta, Abhradeep. Private convex empirical risk minimization and high-dimensional regression. In *COLT*, 2012.

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

NIH. The international hapmap project. *Nature*, 426: 789–796, 2003.

Pathak, Manas A., Rane, Shantanu, and Raj, Bhiksha. Multiparty differential privacy via aggregation of locally trained classifiers. In *NIPS*, pp. 1876–1884, 2010.

Rubinstein, Benjamin I. P., Bartlett, Peter L., Huang, Ling, and Taft, Nina. Learning in a large function space: Privacy-preserving mechanisms for svm learning. *CoRR*, abs/0911.5708, 2009.

Shalev-Shwartz, Shai, Shamir, Ohad, Srebro, Nathan, and Sridharan, Karthik. Stochastic Convex Optimization. In *Proceedings of the Conference on Learning Theory (COLT)*, 2009.

Williams, Oliver and McSherry, Frank. Probabilistic inference and differential privacy. In *NIPS*, pp. 2451–2459, 2010.

A. Interactive Model

A.1. Privacy Guarantee

We restate a version of the privacy theorem by (Gupta et al., 2011) in the context of this paper.

Theorem 9 (Theorem 4.1 from Gupta et al. (2011))**.** *Let T be the total number of queries and B be the number of updates allowed in Algorithm 1, let $\epsilon_0 = \frac{\epsilon}{200\sqrt{B}S\log(4/\delta)}$ and $\sigma = \frac{4}{\epsilon_0}\log(2T/\beta)$, where S is the maximum change in the output of a query (using $\mathbf{w}^*$) when any one entry in the underlying data set is arbitrarily modified. Let (ϵ, δ) be the privacy parameters and β be the failure probability in Algorithm 1. Under this setting, Algorithm 1 is (ϵ, δ)-differentially private.*

We now provide privacy proof of our PINP algorithm (Algorithm 1).

Proof of Theorem 2. The proof proceeds in two stages. In the first stage, we show that prediction function is relatively insensitive to change in the dataset. Specifically, we bound $|\left\langle \mathbf{w}^*_{\mathcal{G}}, \phi(\mathbf{z})\right\rangle - \left\langle \mathbf{w}^*_{\mathcal{G}'}, \phi(\mathbf{z})\right\rangle|$, where $\mathbf{z} \in \mathcal{X}$ and $\mathcal{G}, \mathcal{G}'$ are two datasets differing in exactly one data point. Here $\mathbf{w}^*_{\mathcal{G}}$ and $\mathbf{w}^*_{\mathcal{G}'}$ represent optimal solution to regularized ERM (2) when the underlying datasets are $\mathcal{G}$ and $\mathcal{G}'$, respectively. In the second stage, we invoke Theorem 9 with sensitive bound $|\left\langle \mathbf{w}^*_{\mathcal{G}}, \phi(\mathbf{z})\right\rangle - \left\langle \mathbf{w}^*_{\mathcal{G}'}, \phi(\mathbf{z})\right\rangle|$ to complete the proof.

W.l.o.g. we can assume that the datasets $\mathcal{G}$ and $\mathcal{G}'$ differ in the n-th data point, i.e., $(\mathbf{x}_n, y_n) \in \mathcal{G}$ and $(\mathbf{x}'_n, y'_n) \in \mathcal{G}'$. Now, using optimality of $\mathbf{w}^*_{\mathcal{G}}$ and $\mathbf{w}^*_{\mathcal{G}'}$ for (2) (with dataset $\mathcal{G}$ and $\mathcal{G}'$ respectively) and strong convexity of the ERM (2):

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n-1}\ell(\left\langle \mathbf{w}^*_{\mathcal{G}'}, \phi(\mathbf{x}_i)\right\rangle; y_i) + \frac{1}{n}\ell(\left\langle \mathbf{w}^*_{\mathcal{G}'}, \phi(\mathbf{x}_n)\right\rangle; y_n) \\ + \frac{\lambda}{2}\|\mathbf{w}^*_{\mathcal{G}'}\|_2^2 \\ \geq \frac{1}{n}\sum_{i=1}^{n-1}\ell(\left\langle \mathbf{w}^*_{\mathcal{G}}, \phi(\mathbf{x}_i)\right\rangle; y_i) + \frac{1}{n}\ell(\left\langle \mathbf{w}^*_{\mathcal{G}}, \phi(\mathbf{x}_n)\right\rangle; y_n) \\ + \frac{\lambda}{2}\|\mathbf{w}^*_{\mathcal{G}}\|_2^2 + \frac{\lambda}{2}\|\mathbf{w}^*_{\mathcal{G}} - \mathbf{w}^*_{\mathcal{G}'}\|_2^2.\end{aligned}$$

Hence,

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n-1}\ell(\left\langle \mathbf{w}^*_{\mathcal{G}}, \phi(\mathbf{x}_i)\right\rangle; y_i) + \frac{1}{n}\ell(\left\langle \mathbf{w}^*_{\mathcal{G}}, \phi(\mathbf{x}'_n)\right\rangle; y'_n) \\ + \frac{\lambda}{2}\|\mathbf{w}^*_{\mathcal{G}}\|_2^2 \\ \geq \frac{1}{n}\sum_{i=1}^{n-1}\ell(\left\langle \mathbf{w}^*_{\mathcal{G}'}, \phi(\mathbf{x}_i)\right\rangle; y_i) + \frac{1}{n}\ell(\left\langle \mathbf{w}^*_{\mathcal{G}'}, \phi(\mathbf{x}'_n)\right\rangle; y'_n) \\ + \frac{\lambda}{2}\|\mathbf{w}^*_{\mathcal{G}'}\|_2^2 + \frac{\lambda}{2}\|\mathbf{w}^*_{\mathcal{G}} - \mathbf{w}^*_{\mathcal{G}'}\|_2^2.\end{aligned}$$

Adding the above two equations and using Lipschitz continuity of ℓ:

$$\|\mathbf{w}^*_{\mathcal{G}} - \mathbf{w}^*_{\mathcal{G}'}\|_2 \leq \frac{2LR_\phi}{\lambda n}. \tag{3}$$

Finally, using Cauchy-Schwarz inequality and the above inequality, we have,

$$|\left\langle \mathbf{w}^*_{\mathcal{G}}, \phi(\mathbf{z})\right\rangle - \left\langle \mathbf{w}^*_{\mathcal{G}'}, \phi(\mathbf{z})\right\rangle| \leq \frac{2LR_\phi^2}{\lambda n}.$$

With this bound in hand, we invoke Theorem 9 (Theorem 4.1 by (Gupta et al., 2011)) to complete the proof. □

A.2. Utility Guarantee

In the following we restate a version of Theorem 5.2 from (Gupta et al., 2011) in the context of this paper. Setting the parameters as in Theorem 3 gives us the desired utility guarantee.

Theorem 10 (Theorem 5.2 from Gupta et al. (2011))**.** *Let T be the total number of queries and B be the number of updates allowed in Algorithm 1, let $\epsilon_0 = \frac{\epsilon}{200\sqrt{B}S\log(4/\delta)}$ and $\sigma = \frac{4}{\epsilon_0}\log(2T/\beta)$, where S is the maximum change in the output of a query (using $\mathbf{w}^*$) when any one entry in the underlying data set is arbitrarily modified. Let (ϵ, δ) be the privacy parameters and β be the failure probability in Algorithm 1. As long as the variable counter in Algorithm 1 is less than B, for each query $\mathbf{z}_t$, with probability at least $1-\beta$, the following is true.*

$$|\hat{\mathbf{v}}_t - \langle \phi(\mathbf{z}_t), \mathbf{w}^*\rangle| = O\left(\frac{S\sqrt{B}\log(1/\delta)\log(T/\beta)}{\epsilon}\right)$$

B. Test Data Dependent Learner (Semi-interactive model)

B.1. Privacy Guarantee of Test Data Dependent Learner

Proof of Theorem 4. From (3), we know that for any two training data sets $\mathcal{G}$ and $\mathcal{G}'$ differing in exactly one entry, the following is true:

$$\|\mathbf{w}^*_{\mathcal{G}} - \mathbf{w}^*_{\mathcal{G}'}\|_2 \leq \frac{2LR_\phi}{\lambda n}.$$

Therefore by Cauchy-Schwarz inequality, for any $\mathbf{z} \in \mathcal{X}$ we have

$$|\left\langle \mathbf{w}^*_{\mathcal{G}}, \phi(\mathbf{z})\right\rangle - \left\langle \mathbf{w}^*_{\mathcal{G}'}, \phi(\mathbf{z})\right\rangle| \leq \frac{2LR_\phi^2}{\lambda n}.$$

Theorem now follows by using the above given bound with the following composition theorem. □

Theorem 11 (Composition Theorem from (Dwork et al., 2010))**.** *Let $\epsilon', \delta' > 0$. The class of ϵ-differentially private mechanisms satisfy (ϵ', δ')-differential privacy under k-fold adaptive composition for:*

$$\epsilon' = \sqrt{2k \log(1/\delta')}\epsilon + k\epsilon(e^{\epsilon} - 1).$$

B.2. Utility Guarantee of Test Data Dependent Learner

Proof of Theorem 5. Let,

$$J(\mathbf{w}) = \frac{1}{T}\sum_{t=1}^{T} \left(\langle \mathbf{w}, \phi(\mathbf{z}_t)\rangle - \langle \mathbf{w}^*, \phi(\mathbf{z}_t)\rangle - b_t\right)^2.$$

Since $\hat{\mathbf{w}} = \arg\min_{\mathbf{w}\in\mathcal{C}} J(\mathbf{w})$ and by assumption $\mathbf{w}^* \in \mathcal{C}$, the following holds:

$$\sum_{t=1}^{T}(\langle \hat{\mathbf{w}}, \phi(\mathbf{z}_t)\rangle - \langle \mathbf{w}^*, \phi(\mathbf{z}_t)\rangle)^2 \leq 2\sum_{t=1}^{T} \langle \hat{\mathbf{w}} - \mathbf{w}^*, \phi(\mathbf{z}_t)\rangle\, b_t.$$

Let $\mathbf{b} = \langle b_1, \cdots, b_T\rangle$. Using Cauchy-Schwarz inequality and the fact that $\|\mathbf{v}\|_1 \leq \sqrt{T}\|\mathbf{v}\|_2$, we get:

$$\sum_{t=1}^{T} |\,\langle \hat{\mathbf{w}}, \phi(\mathbf{z}_t)\rangle - \langle \mathbf{w}^*, \phi(\mathbf{z}_t)\rangle\,| \leq 2\sqrt{T}\|\mathbf{b}\|_2.$$

Since ν is the scaling parameter for the Laplace distribution from which each b_t are drawn, therefore by the tail property of Laplace distribution it follows that w.p. $\geq 1-\beta$,

$$\sum_{t=1}^{T} |\,\langle \hat{\mathbf{w}}, \phi(\mathbf{z}_t)\rangle - \langle \mathbf{w}^*, \phi(\mathbf{z}_t)\rangle\,| \leq 2\sqrt{2}T\nu \log(T/\beta)$$

Plugging in the value of $\nu = O\left(\frac{L R_\phi{}^2 \sqrt{T \log(1/\delta)}}{\lambda n \epsilon}\right)$, we have

$$\sum_{t=1}^{T} |\,\langle \hat{\mathbf{w}}, \phi(\mathbf{z}_t)\rangle - \langle \mathbf{w}^*, \phi(\mathbf{z}_t)\rangle\,| = O\left(\frac{T^{3/2} L R_\phi^2 \log(T/\beta)\sqrt{\log(1/\delta)}}{n\epsilon\lambda}\right). \quad (4)$$

Now, define $g(\mathbf{w}; \mathbf{z}_t) = |\,\langle \mathbf{w} - \mathbf{w}^*, \phi(\mathbf{z}_t)\rangle\,|$; note that $g(\mathbf{w}; \mathbf{z}_t)$ is a convex cost functions in $\mathbf{w}$. Now, using Theorem 1 from (Shalev-Shwartz et al., 2009) (stated below) we obtain the following:.

$$\mathbb{E}_{\mathbf{z}\sim\mathcal{P}}\left[g(\hat{\mathbf{w}}; \mathbf{z})\right] \leq \frac{1}{T}\sum_{t=1}^{T} |g(\hat{\mathbf{w}}; \mathbf{z}_t)| + O\left(\frac{\|\mathcal{C}\|_2 R_\phi \sqrt{\log(1/\beta)}}{\sqrt{T}}\right). \quad (5)$$

Therefore, using (4) and (5), we get (w.p. $\geq 1-\beta$):

$$\mathbb{E}_{\mathbf{z}\sim\mathcal{P}}\left[g(\hat{\mathbf{w}}; \mathbf{z})\right] \leq \frac{C_1\sqrt{T} L R_\phi^2 \log(T/\beta)\sqrt{\log(1/\delta)}}{n\epsilon\lambda} + \frac{C_2\|\mathcal{C}\|_2 R_\phi\sqrt{\log(1/\beta)}}{\sqrt{T}},$$

where $C_1, C_2 > 0$ are global constants.

Theorem now follows by setting T as mentioned in the theorem along with using Lipschitz property of ℓ. $\square$

Theorem 12 (Theorem 1 from (Shalev-Shwartz et al., 2009))**.** *Let $\mathcal{C} = \{\mathbf{w} : \|\mathbf{w}\|_2 \leq B\}$ be a convex set, let $\phi : \mathcal{X} \to \mathbb{R}^{d_\phi}$ be a feature map with the image of ϕ has L_2-norm of at most R_ϕ, and let $f : \mathbb{R}\times\mathcal{X} \to \mathbb{R}$ be a L_f-Lipschitz continuous convex cost function in its first parameter. Then for any $\mathcal{P}$ over the domain $\mathcal{X}$, and for $Z = \{\mathbf{z}_1, \cdots, \mathbf{z}_T\}$ drawn i.i.d. from $\mathcal{P}$, the following is true with probability at least $1-\beta$.*

$$\sup_{\mathbf{w}\in\mathcal{C}}\left|\mathbb{E}_{\mathbf{z}\sim\mathcal{P}}\left[f(\langle \mathbf{w}, \phi(\mathbf{z})\rangle; \mathbf{z})\right] - \frac{1}{T}\sum_{t=1}^{T}\left[f(\langle \mathbf{w}, \phi(\mathbf{z}_t)\rangle; \mathbf{z}_t)\right]\right| \leq O\left(\sqrt{\frac{B^2(R_\phi L_f)^2\log(1/\beta)}{T}}\right)$$

B.3. Generalization Bound for Test Data Dependent Learner

Theorem 13 (Error Bound over Test Distribution)**.** *Let $\mathcal{P}$ be a fixed test distribution and let $Z = \{\mathbf{z}_1, \cdots, \mathbf{z}_T\}$ be sampled uniformly from $\mathcal{P}$. If $T = O\left(\frac{\|\mathcal{C}\|_2 n\epsilon\lambda}{L R_\phi\sqrt{\log(1/\delta)}}\right)$ and $\mathbf{w}^* \in \mathcal{C}$ in Algorithm 2, then w.p. $1-\beta$,*

$$\mathbb{E}_{\mathbf{z}\sim\mathcal{P}}[\ell(\langle \widehat{\mathbf{w}}, \phi(q_i)\rangle; y_{q_i})] = \mathbb{E}_{\mathbf{z}\sim\mathcal{P}}[\ell(\langle \mathbf{w}^*, \phi(q_i)\rangle; y_{q_i})] + O\left(\frac{(L R_\phi)^{3/2}\sqrt{\|\mathcal{C}\|_2 \log^{1/2}(1/\delta)}\log(T/\beta)}{\sqrt{n\epsilon\lambda}}\right).$$

C. Test Data Independent Learner (Non-interactive model)

Proof sketch of Theorem 7. For a given dataset $\mathcal{G}$, let $f(\mathcal{G}) = \left(\frac{\epsilon_0 n\lambda}{8 L R_\phi^2}|\,\langle \phi(\mathbf{z}), \mathbf{w}_t - \mathbf{w}^*(\mathcal{G})\rangle\,|\right)$. Using the fact that $\|\mathbf{w}^*(\mathcal{G}) - \mathbf{w}^*(\mathcal{G}')\|_2 \leq \frac{2 L R_\phi}{n\lambda}$ for any two datasets $\mathcal{G}$ and $\mathcal{G}'$ differing in exactly one entry (see Theorem 2 from Section 5), it directly follows that $|f(\mathcal{G}) - f(\mathcal{G}')| \leq \frac{\epsilon_0}{4}$. Hence, it follows that each iteration of Line 3 in Algorithm 3 is $\epsilon_0/2$-differentially private. Now from the analysis of Theorem 2 (from Section 5), it follows that Algorithm 3 is (ϵ, δ)-differentially private. $\square$

Proof of Theorem 8. **Intuition:** The proof of this theorem goes via the following key insight: if we can make almost every round of Algorithm 3 an update round, then the iterates $\mathbf{w}_t$ will become representative of $\mathbf{w}^*$ as time t progresses. This can be formalized via a simple potential argument. (See (Gupta et al., 2011) for the exact formalization.) The way we ensure that each iteration is an update round is by finding a $\mathbf{z}$ (via exponential mechanism) such that it can distinguish between $\hat{\mathbf{w}}_t$ and $\mathbf{w}^*$ with high probability, (*i.e.*, the value of $\langle \phi(\mathbf{z}), \hat{\mathbf{w}} - \mathbf{w}^* \rangle$ is greater than $\frac{\sigma}{4}$).

Main Proof: We apply exponential mechanism to a finite set $S = \{\mathbf{z} : \mathbf{z} \text{ is the center of the } \nu\text{-net}\}$, where ν is as given in the Theorem. That is, we divide the entire space into (overlapping) L_2 balls of radius ν and S is the collection of centers of all such balls. Also, it is known that $|S| = \left(\frac{4}{\nu}\right)^d$.

Now, using the exponential distribution specified in Step 3 of Algorithm 3, we get:

$$\Pr[\ \mathbf{z} \text{ s.t. } |\langle \phi(\mathbf{z}), \mathbf{w}_t - \mathbf{w}^* \rangle| \le OPT_\nu - \gamma] \le |S| e^{-\Lambda \gamma},$$

where $OPT_\nu = \max_{\mathbf{z} \in S} |\langle \phi(\mathbf{z}), \mathbf{w}_t - \mathbf{w}^* \rangle|$ and $\Lambda = \frac{\epsilon_0 n \lambda}{8 L R_\phi^2}$. Hence, w.p. at least $1-\beta$, a $\mathbf{z}$ is sampled s.t.,

$$|\langle \phi(\mathbf{z}), \mathbf{w}_t - \mathbf{w}^* \rangle| \ge OPT_\nu - \frac{\ln(|S|/\beta)}{\Lambda}.$$

Now, let OPT^* be the maximum value of $|\langle \phi(\mathbf{z}), \mathbf{w}_t - \mathbf{w}^* \rangle|$ over the input space $\mathcal{X}$, i.e., $OPT^* = \max_{\mathbf{z} \in \mathcal{X}} |\langle \phi(\mathbf{z}), \mathbf{w}_t - \mathbf{w}^* \rangle|$. Also, $\|\mathbf{z}^* - \mathbf{z}_\nu\|_2 \le 2\nu$ where $z_\nu = \arg\max_{\mathbf{z} \in S} |\langle \phi(\mathbf{z}), \mathbf{w}_t - \mathbf{w}^* \rangle|$ is the optimal over S. Hence, using Lipschitz continuity of the mapping ϕ, we obtain a sample $\mathbf{z}$ w.p. at least $1-\beta$ s.t.:

$$|\langle \phi(\mathbf{z}), \mathbf{w}_t - \mathbf{w}^* \rangle| \ge OPT^* - \frac{\ln(|S|/\beta)}{\Lambda} - \frac{2\nu L_\phi R_\phi L}{\lambda}.$$

Hence, selecting $\nu = \frac{d R_\phi}{\epsilon_0 n L_\phi}$, we get $|\langle \phi(\mathbf{z}), \mathbf{w}_t - \mathbf{w}^* \rangle| \ge OPT^* - \Omega\left(\frac{d L R_\phi^2 \ln(L_\phi) \ln(1/\beta)}{\lambda \epsilon_0 n}\right)$. Now, using Theorem 7.3 of (Gupta et al., 2011) (see Theorem 14), we get,

$$\begin{aligned} &|\langle \phi(\mathbf{z}), \mathbf{w}_t - \mathbf{w}^* \rangle| \le \sigma \\ &\quad = \max\left(\frac{\|\mathbf{w}^*\| R_\phi}{2\sigma\epsilon}, \frac{d L^2 R_\phi^6 \ln(L_\phi) \ln\frac{1}{\beta} \|\mathbf{w}^*\|^2}{\sigma^2 \lambda^2 n^2 \epsilon}\right), \end{aligned}$$

for all $\mathbf{z} \in \mathbb{R}^d$ and $\|\mathbf{z}\|_2 \le 1$. Hence, minimizing over σ, we get

$$\begin{aligned} &|\langle \phi(\mathbf{z}), \mathbf{w}_t - \mathbf{w}^* \rangle| \\ &\quad = O\left(\frac{\|\mathbf{w}^*\| R_\phi^2 d^{1/3} L^{2/3} \ln L_\phi \log^2 1/\delta \ln(1/\beta)}{(\lambda n)^{2/3} \sqrt{\epsilon}}\right), \end{aligned}$$

for all $\mathbf{z} \in \mathbb{R}^d$ and $\|\mathbf{z}\|_2 \le 1$. The theorem now follows using Lipschitz continuity of the loss function ℓ and using the bound $\|\mathbf{w}^*\|_2 \le 2 L R_\phi / \lambda$.

Theorem 14 (Modified Theorem 7.3 from (Gupta et al., 2011))**.** *If the distinguisher in Line 3 of Algorithm 3 outputs a* $\mathbf{z}$ *(with* $\|\mathbf{z}\|_2 \le 1$*) at each step* $t \in \{1, \cdots, B\}$ *such that with probability at least* $1-\beta$ *(over all the* B*-steps),* $|\langle \mathbf{w}^* - \mathbf{w}_t, \phi(\mathbf{z}) \rangle| = \max_{\mathbf{z}_1 \in \mathcal{X}, \|\mathbf{z}_1\|_2 \le 1} |\langle \mathbf{w}^* - \mathbf{w}_t, \phi(\mathbf{z}_1) \rangle| - \Omega\left(\frac{d L R_\phi^2 \ln(L_\phi) \ln(1/\beta)}{\lambda \epsilon_0 n}\right)$*, then for all* $\mathbf{z} \in \mathbb{R}^d$ *with* $\|\mathbf{z}\|_2 \le 1$*, with probability at least* $1-\beta$ *(over all the* B*-steps),* $|\langle \phi(\mathbf{z}), \mathbf{w}_t - \mathbf{w}^* \rangle| \le \mu$*, where* $\mu = \max\left(\frac{\|\mathbf{w}^*\| R_\phi}{2\sigma\epsilon}, \frac{d L^2 R_\phi^6 \ln(L_\phi) \ln\frac{1}{\beta} \|\mathbf{w}^*\|^2}{\sigma^2 \lambda^2 n^2 \epsilon}\right)$.

□