Active Local Learning

Arturs Backurs BACKURS@TTIC.EDU
Toyota Technological Institute at Chicago

Avrim Blum AVRIM@TTIC.EDU
Toyota Technological Institute at Chicago

Neha Gupta NEHAGUPTA@CS.STANFORD.EDU
Stanford University

Editors: Jacob Abernethy and Shivani Agarwal

Abstract

In this work we consider active *local learning*: given a query point x, and active access to an unlabeled training set S, output the prediction $h(x)$ of a near-optimal $h \in H$ using significantly fewer labels than would be needed to actually learn h fully. In particular, the number of label queries should be independent of the complexity of H, and the function h should be well-defined, independent of x. This immediately also implies an algorithm for *distance estimation*: estimating the value $opt(H)$ from many fewer labels than needed to actually learn a near-optimal $h \in H$, by running local learning on a few random query points and computing the average error.

For the hypothesis class consisting of functions supported on the interval $[0, 1]$ with Lipschitz constant bounded by L, we present an algorithm that makes $O((1/\epsilon^6)\log(1/\epsilon))$ label queries from an unlabeled pool of $O((L/\epsilon^4)\log(1/\epsilon))$ samples. It estimates the distance to the best hypothesis in the class to an additive error of ϵ for an arbitrary underlying distribution. We further generalize our algorithm to more than one dimensions. We emphasize that the number of labels used is independent of the complexity of the hypothesis class which is linear in L in the one-dimensional case. Furthermore, we give an algorithm to locally estimate the values of a near-optimal function at a few query points of interest with number of labels independent of L.

We also consider the related problem of approximating the minimum error that can be achieved by the Nadaraya-Watson estimator under a linear diagonal transformation with eigenvalues coming from a small range. For a d-dimensional pointset of size N, our algorithm achieves an additive approximation of ϵ, makes $\tilde{O}(d/\epsilon^2)$ queries and runs in $\tilde{O}(d^2/\epsilon^{d+4} + dN/\epsilon^2)$ time.

Keywords: Local Learning, Error Estimation, Lipschitz functions, Nadaraya-Watson estimator

1. Introduction

Consider a setting where we have a large amount of unlabeled data but the corresponding labels are expensive to obtain. Our aim is to understand what information can we reliably obtain about the predictor that we would have learned had we been given unlimited labeled data by actively querying only a few labels (“few” = independent of the complexity of the hypothesis class).

In particular, we look at the following two related questions. First, given an unlabeled set of training data sampled from a distribution, is it possible to estimate how well the best prediction function in a hypothesis class would do when the number of labels that we can actually obtain is insufficient to learn the best prediction function accurately? Second, suppose we have a few queries of interest of which we are interested in the labels of, is it possible to output the predictions for those queries corresponding to a nearly optimal function in the class without running the full training algorithm. Note that a nearly optimal function corresponds to a function which has low total error on data with respect to the underlying distribution. There are many natural scenarios in which this could be useful. For example, consider a setting where we are interested in predicting the outcome of a treatment for a particular patient with a certain disease. Medical records of patients who had similar treatments in the past are available in a hospital database but since the treatment outcome is sensitive and private, we want to minimize the number of patients for whom we acquire the information. Note that in this case, we cannot directly acquire the label of the query patient because the treatment procedure is an intervention which cannot be undone. Formally, we are interested in computing the labels of a few particular queries "locally" and "parallelly" with few label queries (independent of the complexity measure of the hypothesis class). We would like to mention that both the aforementioned problems are related because if we have an algorithm for predicting the labels of a few test points with few label queries, we can sample a few data points and use their labels to estimate the error of the best function in the class. While the question of error estimation has been previously studied in certain settings (Kong and Valiant, 2018; Blum and Hu, 2018), there has been no prior work for local and parallel predictions for a function class with global constraints in learning based settings to the best of our knowledge.

In this work, we answer these questions in affirmative. For the class of Lipschitz functions with Lipschitz constant at most L, we show that it is possible to estimate the error of the best function in the class with respect to an underlying distribution with independent of L label queries. We also show that it is possible to locally estimate the values of a nearly optimal function at a few test points of interest with independent of L label queries. A key point to notice here is that a function can predict any arbitrary values for the fixed constant number of queries and still be approximately optimal in terms of the total error since the queries have a zero measure with respect to the underlying distribution. Therefore, the additional guarantee that we have is after a common preprocessing step, the combined function, obtained when the algorithm is run in parallel for all possible query points independently, is L-Lipschitz and approximately optimal in terms of total error with respect to the underlying distribution.

We also give an algorithm for estimating the minimum error for Nadaraya-Watson prediction algorithm amongst a set of linear transformations which is efficient in terms of both sample and time complexity. Note that computing the prediction for even a single query point requires computing a weighted sum of the labels of the training points and hence requires Nd time and N labels where N is the number of points in the training set and d is the dimension of the data points. Moreover, naively computing the error for a set of linear transformations with size exponential in d would require exponential in d labels and time which has a multiplicative factor of N and $1/\epsilon^d$ where ϵ is the accuracy parameter. However, we obtain sample complexity which only depends polynomially on d and logarithmically on N and time complexity without the multiplicative dependence of N and $1/\epsilon^d$.

We would like to clarify how the setting considered in this paper differs from the classical notion of local learning algorithms. The notion of local learning algorithms (Bottou and Vapnik, 1992) is used to refer to learning schemes where the prediction at a point depends locally on the training points around it. The key distinction is that rather than proposing a local learning strategy and seeing how well it performs, we are looking at the question of whether we can find local algorithms for simulating empirical risk minimization for a hypothesis class with global constraints.

The remainder of the paper is organized as follows. We begin in Section 2 with a high level overview of our results and discuss the related work in Section 3. Section 4 is devoted to our main results for the class of one dimensional Lipschitz functions. In Section 5, we present our results for error estimation for Nadaraya-Watson estimator. Finally, we end with a conclusion and some possible future directions in Section 6.

2. Our Results

Consider a class of one dimensional Lipschitz functions supported on the interval $[0, 1]$ with Lipschitz constant at most L and data drawn from an arbitrary unknown distribution $\mathcal{D}$ with labels in the range $[0, 1]$. For this setting, we show (Theorem 1) that there exists a function $\tilde{f}$ in the class that is optimal up to an additive error of ϵ such that for any query x, $\tilde{f}(x)$ can be computed with $O((1/\epsilon^4)\log(1/\epsilon))$ label queries to a pool of $O((L/\epsilon^4)\log(1/\epsilon))$ unlabeled samples drawn from the distribution $\mathcal{D}$. Also, the function values $\tilde{f}(x)$ at these query points x can be computed in parallel once the unlabeled random samples have been drawn and fixed beforehand. Note that standard empirical risk minimization approaches would require a sample complexity of $O(L/\epsilon^3)$ to output the value of an approximately optimal function even at a single query point. But, the number of samples required by our approach for constant number of queries is independent of L (which determines the complexity of the function class required for learning). In this setting, we think of L as large compared to ϵ which is the error parameter. At a high level, we show that it is possible to effectively reduce the hypothesis class of bounded Lipschitz functions to a strictly smaller class of piece-wise independent Lipschitz functions where the function value can be computed locally by not losing too much in terms of the total accuracy.

We also show (Theorem 2) that for the class of L Lipschitz functions considered above, it is possible to estimate the error of the optimal function in the class up to an additive error ϵ using $O((1/\epsilon^6)\log(1/\epsilon))$ active label queries over an unlabeled pool of size $O((L/\epsilon^4)\log(1/\epsilon))$. The idea is to compute the empirical error of the local function $\tilde{f}(x)$ constructed above by using $O(1/\epsilon^2)$ random samples from distribution D. Since $\tilde{f}(x)$ can be computed locally for a given query x, the total number of labels needed is independent of L. Using standard concentration results and the fact that $\tilde{f}$ is ϵ-optimal, we get the desired estimate.

We also extend the results to the case of more than one dimensions where the dimension is constant with respect to the Lipschitz constant L. The results are mentioned in Theorems 14 and 15.

For the related setting of Nadaraya-Watson estimator, we show (Theorem 4) that it is possible to estimate the minimum error that can be achieved under a linear diagonal transformation with eigenvalues bounded in a small range with additive error at most ϵ by making $\tilde{O}(d/\epsilon^2)$ label queries over a d-dimension unlabeled training set with size N in running time $\tilde{O}(d^2/\epsilon^{d+4} + dN/\epsilon^2)$. Note that exactly computing the prediction for even a single data point requires going over the entire dataset thus needing N label queries and Nd time. Moreover, naively computing the error for each of the linear diagonal transformation with bounded eigenvalues would require number of labels depending on $1/\epsilon^d$ and a running time depending multiplicatively on N and $1/\epsilon^d$. In comparison, we achieve a labeled query complexity independent of N and polynomial dependence on the dimension d. Moreover, we separate the multiplicative dependence of N and $1/\epsilon^d$ in the running time to an additive dependence. We will further elaborate on our algorithm and the comparison with the standard algorithm in Section 5.

3. Related Work

Local Computation Algorithms. Our work on locally learning Lipschitz functions closely resembles the concept of local computation algorithms introduced in the pioneering work of Rubinfeld et al. (2011). They were interested in the question of whether it is possible to compute specific parts of the output in time sublinear in the input size. As mentioned by the authors, that work was a formalization of different forms of this concept already existing in literature in the form of local distributed computation, local algorithms, locally decodable codes and local reconstruction models. The reader can refer to Rubinfeld et al. (2011) for the detailed discussion of work in each of these subfields. In the paper, the authors looked at several graph problems in the local computation model like maximal independent set, k-SAT and hypergraph coloring. Since then, there has been a lot of further work on local computation algorithms for various graph problems including maximum matching, load balancing, set cover and many others (Mansour and Vardi (2013); Alon et al. (2012); Mansour et al. (2012); Parnas and Ron (2007); Parter et al. (2019); Levi et al. (2014); Grunau et al. (2020)). There also has been work on solving linear systems locally in sublinear time (Andoni et al., 2018) for the special case of sparse symmetric diagonally dominant matrices with small condition number. However, computing the best Lipschitz function for a given set of data cannot be written as a linear system. Moreover, the primary focus of all of these works has been on sublinear computational complexity, whereas we focus primarily on sample complexity.

In another related work, Mansour et al. (2014) used local computation algorithms in the context of robust inference to give polynomial time algorithms. They formulated their inference problem as an exponentially sized linear program and showed that the linear program (LP) has a special structure which allowed them to compute the values of certain variables in the optimal solution of the linear program in time sublinear in the total number of variables. They did this by sampling a polynomial number of constraints in the LP. Note that for our setting for learning Lipschitz functions, given all the unlabeled samples, learning the value of the best Lipschitz function on a particular input query can be cast as a linear program. However, our LP does not belong to the special class of programs that they consider. Moreover, we have a continuous domain and the number of possible queries is infinite. We cannot hope to get a globally Lipschitz solution by locally solving a smaller LP with constraints sampled independently for each query. We have to carefully design the local intervals and use different learning strategies for different types of intervals to ensure that the learned function is globally Lipschitz and also has good error bounds.

In another work, Feige et al. (2015) considered the use of local computation algorithms for inference settings. They reduced their problem of inference for a particular query to the problem of computing minimum vertex cover in a bipartite graph. However, the focus was on time complexity rather than sample complexity. The problem of computing a part of output of the learned function in sublinear sample complexity as compared to the usual notion of complexity of the function class (VC-dimension, covering numbers) has not been previously looked at in the literature to the best of our knowledge.

Transductive Inference. Another related line of work that has been done in the learning theory community is based on transductive inference (Vapnik, 1998) which as opposed to inductive inference aims to estimate the values of the function on a few given data points of interest. The philosophy behind this line of work is to avoid solving a more general problem as an intermediate step to solve a problem. This idea is very similar to the idea considered here wherein we are interested in computing the prediction at specific query points of interest. However, our prediction algorithm still requires a guarantee on the total error over the complete domain with respect to the underlying distribution though it is never explicitly constructed for the entire domain unless required. The reader can refer to chapters 24 and 25 in Chapelle et al. (2009) for additional discussion on the topic.

Local Learning Algorithms. The term local learning algorithms has been used to refer to the class of learning schemes where the prediction at a test point depends only on the training points in the vicinity of the point such as k-nearest neighbor schemes (Bottou and Vapnik, 1992; Vapnik, 1992). However, these works have primarily focused on proposing different local learning strategies and evaluating how well they perform. In contrast, we are interested in the question of whether local algorithms can be used for simulating empirical risk minimization for a hypothesis class with global constraints such as the class of Lipschitz functions.

Property Testing. There also has been a large body of work in the theoretical computer science community on property testing where the goal is to determine whether a function belongs to a class of functions or is far away from it with respect to some notion of distance in sublinear sample complexity. The commonly studied testing problems include testing monotonicity of a function over a boolean hypercube (Goldreich et al., 1998b; Chakrabarty and Seshadhri, 2016; Chen et al., 2014; Khot et al., 2018), testing linearity over finite fields (Bellare et al., 1996; Ben-Sasson et al., 2003), testing for concise DNF formulas (Diakonikolas et al., 2007; Parnas et al., 2002) and testing for small decision trees (Diakonikolas et al., 2007). However, all the aforementioned algorithms work in a query model where a query can be made on any arbitrary domain point of choice.

The setting which is closer to learning where the labels can only be obtained from a fixed distribution was first studied by Goldreich et al. (1998a); Kearns and Ron (2000). This setting is also called as passive property testing. The notion of active property testing was first introduced by Balcan et al. (2012) where an algorithm can make active label queries on an unlabeled sample of points drawn from an unknown distribution. However, one limitation of these algorithms is that they do not give meaningful bounds to distinguish the function from being approximately close to the function class (rather than belonging to the class) vs. far away from it.

Parnas et al. (2006) first introduced the notion of tolerant testing where the aim is to detect whether the function is ϵ close to the class or 2ϵ far from it in the query model with queries on arbitrary domains points of choice. This also relates to estimating the distance of the function from the class within an additive error of ϵ. Blum and Hu (2018) first studied the problem of active tolerant testing where they were interested in algorithms which are tolerant to the function not being exactly in the function class and also have active query access to the labels over the unlabeled samples from an unknown distribution. Specifically, Blum and Hu (2018) gave algorithms for estimating the distance of the function from the class of union of d intervals with a labeled sample complexity of $\text{poly}(1/\epsilon)$. The key point to be noted is that the labeled sample complexity is independent of d which is the VC dimension of that class and dictates the number of samples required for learning. We note that our algorithm for error estimation for the class of Lipschitz functions in labeled sample complexity independent of L is another work along these lines.

Property testing has also been studied for the specific class of Lipschitz functions (Jha and Raskhodnikova, 2013; Chakrabarty and Seshadhri, 2013; Berman et al., 2014). However, all of these results either query arbitrary domain points or are in the non-tolerant setting or work only for discrete domain.

A closely related work to our method of error estimation where the predicted label depends on the labels of the training data weighted according to some appropriate kernel function was also considered in Blum and Hu (2018). In particular, they looked at the setting where the predicted label is based on the k-nearest neighbors and showed that the ℓ_1 loss can be estimated up to an additive error of ϵ using $O(1/\epsilon^2)$ label queries on $N + O(1/\epsilon^2)$ unlabeled samples. Their results also extend to the case where the prediction is the weighted average of all the unlabeled points in the sample by sampling with a probability proportional to the weight of the point. However, this sampling when repeated for $O(1/\epsilon^d)$ different linear transformations would lead to a labeled sample complexity depending on $1/\epsilon^d$. Moreover, sampling a point with a probability proportional to the weight would require a running time of N per query and thus, would give a total running time of $O(N/\epsilon^{d+2})$.

4. Lipschitz Functions

In this section, we describe the formal problem setup for both local learning and error estimation for the class of Lipschitz functions. We then proceed to describe our algorithms and derive label query complexity bounds for them. We restrict our attention to the one-dimensional problems here and defer the details of the more than one dimensional setup to Appendix B.

4.1. Problem Setup

For any fixed $L > 0$, let $\mathcal{F}_L$ be the class of d dimensional functions supported on the domain $[0,1]^d$ with Lipschitz constant at most L, that is,

$$\mathcal{F}_L = \{f : [0,1]^d \mapsto [0,1],\ f \text{ is } L\text{-Lipschitz }\}. \tag{1}$$

Let $\mathcal{D}$ be any distribution over $[0,1]^d \times [0,1]$ and $\mathcal{D}_x$ be the corresponding marginal distribution over $[0,1]^d$. The prediction error of a function f with respect to $\mathcal{D}$ and the optimal prediction error of the class $\mathcal{F}_L$ are

$$\mathrm{err}_{\mathcal{D}}(f) := E_{x,y\sim\mathcal{D}}|y - f(x)| \quad \text{and} \quad \mathrm{err}_{\mathcal{D}}(\mathcal{F}_L) := \min_{f\in\mathcal{F}_L} E_{x,y\sim\mathcal{D}}|y - f(x)|.$$

Also, let us denote the error of a function f relative to another function f' and function class $\mathcal{F}_L$ as

$$\Delta_{\mathcal{D}}(f, f') := \mathrm{err}_{\mathcal{D}}(f) - \mathrm{err}_{\mathcal{D}}(f') \quad \text{and} \quad \Delta_{\mathcal{D}}(f, \mathcal{F}_L) := \mathrm{err}_{\mathcal{D}}(f) - \mathrm{err}_{\mathcal{D}}(\mathcal{F}_L). \tag{2}$$

We say that a function $f \in \mathcal{F}_L$ is ϵ-optimal with respect to distribution $\mathcal{D}$ and the function class $\mathcal{F}_L$ if it satisfies $\Delta_{\mathcal{D}}(f, \mathcal{F}_L) \le \epsilon$. Let functions $f^*_{\mathcal{D}} \in \mathcal{F}_L$ and $\hat{f}_S \in \mathcal{F}_L$ be the minimizers of error with respect to the distribution and empirical error on set S defined as

$$f^*_{\mathcal{D}} = \operatorname*{argmin}_{f\in\mathcal{F}_L} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} |y - f(x)| \quad \text{and} \quad \hat{f}_S = \operatorname*{argmin}_{f\in\mathcal{F}_L} \sum_{(x_j,y_j)\in S} |y_j - f(x_j)|. \tag{3}$$

Local Learning. Given access to unlabeled samples from $\mathcal{D}_x$ and a test point x^*, the objective of *Local Learning* is to output the prediction $\tilde{f}(x^*)$ using a small number of label queries such that $\tilde{f} \in \mathcal{F}_L$ is ϵ-optimal. In addition, we would like such a predictor, Alg, to be able to answer multiple such queries while being consistent with the same function $\tilde{f}$, that is,

$$\mathrm{Alg}(x^*) = \tilde{f}(x^*) \quad \text{for all } x^* \in [0,1]^d.$$

Error Estimation. Given access to unlabeled samples from distribution $\mathcal{D}_x$, the goal of *Error Estimation* is to output an estimate of the optimal prediction error $\mathrm{err}_{\mathcal{D}}(\mathcal{F}_L)$ up to an additive error of ϵ using few label queries.

Algorithm 1 Preprocess($L, \mathcal{D}_x, \epsilon$)

```
Sample a uniformly random offset $b_1$ from $\{1, 2, \cdots, \frac{1}{\epsilon}\}\frac{1}{L}$.
Divide the $[0, 1]$ interval into alternating intervals of length $\frac{1}{L\epsilon}$ and $\frac{1}{L}$ with boundary at $b_1$ and let $\mathcal{P}$ be the resulting partition, that is, $\mathcal{P} = \{[b_0 = 0, b_1], [b_1, b_2], \ldots, \}$ where $b_2 = b_1 + \frac{1}{L}, b_3 = b_2 + \frac{1}{L\epsilon}, \ldots$.
Sample a set $S = \{x_i\}_{i=1}^{M}$ of $M = O(\frac{L}{\epsilon^4}\log(\frac{1}{\epsilon}))$ unlabeled examples from distribution $\mathcal{D}_x$.
**Output** $S, \mathcal{P}$.
```

4.2. Guarantees for Local Learning

We begin by describing our proposed algorithm for Local Learning and then provide a bound on its query complexity in Theorem 2.

Our algorithm for local predictions first involves a preprocessing step (Algorithm 1) which takes as input the Lipschitz constant L, sampling access to distribution $\mathcal{D}_x$ and the error parameter ϵ and returns a partition $\mathcal{P} = \{I_1 = [b_0, b_1], I_2 = [b_1, b_2], \ldots, \}$ and a set S of unlabeled samples. The partition $\mathcal{P}$ consists of alternating intervals of length $1/L$ and $1/(L\epsilon)$ over the domain $[0, 1]$. Let us divide these intervals[1] further into the two sets

$$\mathcal{P}_{\mathsf{lg}} := \{[b_0, b_1], [b_2, b_3], \ldots, \} \text{ (long intervals) and } \mathcal{P}_{\mathsf{sh}} := \{[b_1, b_2], [b_3, b_4], \ldots, \} \text{ (short intervals).}$$

The Query algorithm (Algorithm 2) for test point x^* takes as input the set S of unlabeled samples and the partition $\mathcal{P}$ returned by the Preprocess algorithm. Note that all subsequent queries use the same partition $\mathcal{P}$ and the same set of unlabeled examples S. The algorithm uses different learning strategies depending on whether x^* belongs to one of the long intervals in $\mathcal{P}_{\mathsf{lg}}$ or short intervals in $\mathcal{P}_{\mathsf{sh}}$. For the long intervals, it outputs the prediction corresponding to the empirical risk minimizer (ERM) function restricted to that interval. Whereas for the short interval, the prediction is made by linearly interpolating the function values at the boundaries with the neighbouring long intervals. This linear interpolation ensures that the overall function is Lipschitz. We bound the expected prediction error of this scheme with respect to class $\mathcal{F}_L$ by separately bounding this error for long and short intervals. For the long intervals, we prove that the ERM has low error by ensuring that each interval contains enough unlabeled samples. On the other hand, we show that the short intervals do not contribute much to the error because of their low probability under the distribution $\mathcal{D}$.

Algorithm 2 Query($x, S, \mathcal{P} = \{[b_0, b_1], [b_1, b_2], [b_2, b_3], \ldots\}$)

```
**if** query $x \in I_i = [b_{i-1}, b_i]$ where $I_i \in \mathcal{P}_{\mathsf{lg}}$ **then**
    Query labels[2]for $x \in S \cap I_i$.
    **Output** $\hat{f}_{S \cap I_i}(x)$.
**else if** query $x \in I_i = [b_{i-1}, b_i]$ where $I_i \in \mathcal{P}_{\mathsf{sh}}$ **then**
    Query labels for $x \in S \cap (I_{i-1} \cup I_{i+1})$.
    **if** $b_{i-1} > 0$ **then** $v_i^l = \hat{f}_{S \cap I_{i-1}}(b_{i-1})$ **else** $v_i^l = \hat{f}_{S \cap I_{i+1}}(b_i)$ **end if**
    **if** $b_{i-1} < 1$ **then** $v_i^u = \hat{f}_{S \cap I_{i+1}}(b_i)$ **else** $v_i^u = \hat{f}_{S \cap I_{i-1}}(b_{i-1})$ **end if**
    **Output** $v_i^l + (x - b_{i-1})\frac{v_i^u - v_i^l}{b_i - b_{i-1}}$.
**end if**
```

[1] Note that long intervals at the boundary could be shorter, but those can be handled similarly. Moreover, if any long interval gets more than $\frac{1}{2\epsilon^4}\log(\frac{1}{\epsilon})$ samples, we discard the future samples which fall into that interval.

Next, we state the number of label queries needed to make local predictions corresponding to the Query algorithm (Algorithm 2).

Theorem 1 *For any distribution $\mathcal{D}$ over $[0,1] \times [0,1]$, Lipschitz constant $L > 0$ and error parameter $\epsilon \in [0,1]$, let $(S, \mathcal{P})$ be the output of (randomized) Algorithm 1 where S is the set of unlabeled samples of size $O(\frac{L}{\epsilon^4}\log(\frac{1}{\epsilon}))$ and $\mathcal{P}$ is a partition of the domain $[0,1]$. Then, there exists a function $\tilde{f} \in \mathcal{F}_L$, such that for all $x \in [0,1]$, Algorithm 2 queries $O(\frac{1}{\epsilon^4}\log(\frac{1}{\epsilon}))$ labels from the set S and outputs Query(x, S, P) satisfying*

$$\textit{Query}(x, S, \mathcal{P}) = \tilde{f}(x)\ , \tag{4}$$

and the function $\tilde{f}$ is ϵ-optimal, that is, $\Delta_{\mathcal{D}}(\tilde{f}, \mathcal{F}_L) \le \epsilon$ with probability greater than $\frac{1}{2}$.

Proof We begin by defining some notation. Let $S = \{x_i\}_{i=1}^M$ be the set of the unlabeled samples and $\mathcal{P} = \{[b_0, b_1], [b_1, b_2], \ldots\}$ be the partition returned by the pre-processing step given by Algorithm 1. We will use y_i to denote the queried label for the datapoint x_i. Let $\mathcal{D}_i$ be the distribution of a random variable $(X, Y) \sim \mathcal{D}$ conditioned on the event $\{X \in I_i\}$. Similarly, let $\mathcal{D}_{\mathsf{lg}}$ and $\mathcal{D}_{\mathsf{sh}}$ be the conditional distribution of $\mathcal{D}$ on intervals belonging to $\mathcal{P}_{\mathsf{lg}}$ and $\mathcal{P}_{\mathsf{sh}}$ respectively. Let p_i denote the probability of a point sampled from distribution $\mathcal{D}$ lying in interval I_i. Going forward, we use the shorthand 'probability of interval I_i' to denote p_i. Let p_{lg} and p_{sh} be the probability of set of long and short intervals respectively. Recall that $f^*_{\mathcal{D}}$ is the function which minimizes $\mathrm{err}_{\mathcal{D}}(f)$ for $f \in \mathcal{F}_L$ and $f^*_{\mathcal{D}_i}$ is the function which minimizes this error with respect to the conditional distribution $\mathcal{D}_i$. Let M_i denote the number of unlabeled samples of S lying in interval I_i. For any interval I_i, let $\hat{f}_{S \cap I_i}$ be the ERM with respect to that interval.

Lipschitzness of $\tilde{f}$. For any interval $I_i = [b_{i-1}, b_i]$, let $v_i^l = \hat{f}_{S \cap I_{i-1}}(b_{i-1})$ be the value of the function with minimum empirical error on the neighbouring interval I_{i-1} at the boundary point b_{i-1} and similarly $v_i^u = \hat{f}_{S \cap I_{i+1}}(b_i)$ be the value of the function $\hat{f}_{S \cap I_{i+1}}$ at the boundary point b_i. Note that if I_i is a boundary interval and therefor I_{i-1} (or I_{i+1}) does not exist, we can define $v_i^l = v_i^u$ (or $v_i^u = v_i^l$). Further, let $f_i^{\mathsf{int}} : I_i \mapsto \mathbb{R}$ be the linear function interpolating from v_i^l to v_i^u,

$$f_i^{\mathsf{int}}(x) = v_i^l + (x - b_{i-1})\frac{v_i^u - v_i^l}{b_i - b_{i-1}}.$$

Note that the Query procedure (Algorithm 2) is designed to output $\tilde{f}(x)$ for each query x where

$$\tilde{f}(x) = \begin{cases} \hat{f}_{S \cap I_i}(x) & \text{if } x \in I_i \text{ for any } I_i \in \mathcal{P}_{\mathsf{lg}} \\ f_i^{\mathsf{int}}(x) & \text{if } x \in I_i \text{ for any } I_i \in \mathcal{P}_{\mathsf{sh}} \end{cases}.$$

Now, it is easy to see that the function $\tilde{f}(x)$ is L-Lipschitz. The function $\hat{f}_{S \cap I_i}$ on each of the long intervals is L-Lipschitz by construction. The function f_i^{int} on each of the short intervals is also L-Lipschitz since the short intervals in $\mathcal{P}_{\mathsf{sh}}$ have length $1/L$ and the label values $y_j \in [0,1]$. Also, the function $\tilde{f}$ is continuous at the boundary of each interval by construction.

[2] One can either think of the label as being fixed for every data point or if it is randomized, we need to use the same label for every datapoint once queried.

Error Guarantees for $\tilde{f}$. Now looking at the error rate of the function $\tilde{f}(x)$ and following a repeated application of tower property of expectation, we get that

$$\begin{aligned}\Delta_{\mathcal{D}}(\tilde{f}, \mathcal{F}_L) &= p_{\mathsf{lg}} \Delta_{\mathcal{D}_{\mathsf{lg}}}(\tilde{f}, f^*_{\mathcal{D}}) + p_{\mathsf{sh}} \Delta_{\mathcal{D}_{\mathsf{sh}}}(\tilde{f}, f^*_{\mathcal{D}}) \\ &= \sum_{i:I_i \in \mathcal{P}_{\mathsf{lg}}} p_i \Delta_{\mathcal{D}_i}(\tilde{f}, f^*_{\mathcal{D}}) + p_{\mathsf{sh}} \Delta_{\mathcal{D}_{\mathsf{sh}}}(\tilde{f}, f^*_{\mathcal{D}}) \qquad (5)\end{aligned}$$

We now bound both terms above to obtain a bound on the total error of the function $\tilde{f}$.

Error for short intervals. The probability of short intervals p_{sh} is small with high probability since the total length of short intervals is ϵ and the intervals are chosen uniformly randomly. More formally, from Lemma 6, we know that with probability at least $1 - \delta$, the probability of short intervals p_{sh} is upper bounded by ϵ/δ. Also, the error for any function f is bounded between $[0, 1]$ since the function's range is $[0, 1]$. Hence, we get that

$$p_{\mathsf{sh}} \Delta_{\mathcal{D}_{\mathcal{P}_{\mathsf{sh}}}}(\tilde{f}, f^*_{\mathcal{D}}) \leq \frac{\epsilon}{\delta} \qquad (6)$$

Error for long intervals: We further divide the long intervals into 3 subtypes:

$$\mathcal{P}_{\mathsf{lg},1} := \left\{ I_i \mid I_i \in \mathcal{P}_{\mathsf{lg}},\ p_i \geq \frac{1}{L},\ M_i \geq \frac{1}{2\epsilon^4} \log\left(\frac{1}{\epsilon}\right) \right\}, \mathcal{P}_{\mathsf{lg},2} := \left\{ I_i \mid I_i \in \mathcal{P}_{\mathsf{lg}},\ p_i \geq \frac{1}{L},\ M_i < \frac{1}{2\epsilon^4} \log\left(\frac{1}{\epsilon}\right) \right\},$$
$$\mathcal{P}_{\mathsf{lg},3} := \left\{ I_i \mid I_i \in \mathcal{P}_{\mathsf{lg}},\ p_i < \frac{1}{L} \right\}.$$

The intervals in both first and second subtypes have large probability p_i with respect to distribution $\mathcal{D}$ but differ in the number of unlabeled samples in S lying in them. Finally, the intervals in third subtype $\mathcal{P}_{\mathsf{lg},3}$ have small probability p_i with respect to distribution $\mathcal{D}$. Now, we can divide the total error of long intervals into error in these subtypes

$$\sum_{i:I_i \in \mathcal{P}_{\mathsf{lg}}} p_i \Delta_{\mathcal{D}_i}(\tilde{f}, f^*_{\mathcal{D}}) = \underbrace{\sum_{i:I_i \in \mathcal{P}_{\mathsf{lg},1}} p_i \Delta_{\mathcal{D}_i}(\tilde{f}, f^*_{\mathcal{D}})}_{E1} + \underbrace{\sum_{i:I_i \in \mathcal{P}_{\mathsf{lg},2}} p_i \Delta_{\mathcal{D}_i}(\tilde{f}, f^*_{\mathcal{D}})}_{E2} + \underbrace{\sum_{i:I_i \in \mathcal{P}_{\mathsf{lg},3}} p_i \Delta_{\mathcal{D}_i}(\tilde{f}, f^*_{\mathcal{D}})}_{E3}. \qquad (7)$$

Now, we will argue about the contribution of each of the three terms above.

Bounding $E3$. Since there are at most $L\epsilon$ long intervals and each of these intervals I_i has probability p_i upper bounded by $1/L$, the total probability combined in these intervals is at most ϵ. Also, in the worst case, the loss can be 1. Hence, we get an upper bound of ϵ on E_3.

Bounding $E2$. From Lemma 5, we know that with failure probability at most δ, these intervals have total probability upper bounded by ϵ/δ. Again, the loss can be 1 in the worst case. Hence, we can get an upper bound of ϵ/δ on E_2.

Bounding $E1$. Let F_i denote the event that $\Delta_{\mathcal{D}_i}(\hat{f}_{S \cap I_i}, f^*_{\mathcal{D}_i}) > \epsilon$. The expected error of intervals I_i in $\mathcal{P}_{\mathsf{lg},1}$ is then

$$\begin{aligned}\mathbb{E}[\sum_{i:I_i \in \mathcal{P}_{\mathsf{lg},1}} p_i \Delta_{\mathcal{D}}(\tilde{f}, f^*_{\mathcal{D}})] &\overset{\text{(i)}}{\leq} \mathbb{E}[\sum_{i:I_i \in \mathcal{P}_{\mathsf{lg},1}} p_i \Delta_{\mathcal{D}_i}(\hat{f}_{S \cap I_i}, f^*_{\mathcal{D}_i})] \\ &= \sum_{i:I_i \in \mathcal{P}_{\mathsf{lg},1}} p_i (\mathbb{E}[\Delta_{\mathcal{D}_i}(\hat{f}_{S \cap I_i}, f^*_{\mathcal{D}_i}) | F_i] \Pr(F_i) + \mathbb{E}[\Delta_{\mathcal{D}_i}(\hat{f}_{S \cap I_i}, f^*_{\mathcal{D}_i}) | \neg F_i] \Pr(\neg F_i)) \\ &\overset{\text{(ii)}}{\leq} \sum_{i:I_i \in \mathcal{P}_{\mathsf{lg},1}} p_i (1 \cdot \epsilon + \epsilon \cdot 1) \leq 2\epsilon,\end{aligned}$$

where step (i) follows by noting that $\tilde{f} = \hat{f}_{S \cap I_i}$ for all long intervals $I_i \in \mathcal{P}_{\mathsf{lg}}$ and that $f^*_{\mathcal{D}_i}(x)$ is the minimizer of the error $\mathrm{err}_{\mathcal{D}_i}(f)$ over all L-Lipschitz functions, and step (ii) follows since $\mathbb{E}[\Delta_{\mathcal{D}_i}(\hat{f}_{S \cap I_i}, f^*_{\mathcal{D}_i}) | \neg F_i] \leq \epsilon$ by the definition of event F_i and $\Pr(F_i) \leq \epsilon$ follows from a standard uniform convergence argument (detailed in Lemma 7). Now, using Markov's inequality, we get that $E_1 \leq 2\epsilon/\delta$ with failure probability at most δ.

Plugging the error bounds obtained in equations (6) and (7) into equation (5) and setting $\delta = \frac{1}{10}$ establishes the required claim.

Label Query Complexity. For any query point x^*, $\tilde{f}(x^*)$ can be computed by querying the labels of the interval in which x^* lies (if x^* lies in a long interval) or the two neighboring intervals of the interval in which x^* lies (if x^* lies in a short interval). Hence, the computation requires $O((1/\epsilon^4)\log(1/\epsilon))$ label queries over the set S of $O((L/\epsilon^4)\log(1/\epsilon))$ unlabeled samples. ■

4.3. Guarantees for Error Estimation

We now study the Error Estimation problem for the class of Lipschitz functions $\mathcal{F}_L$. Our proposed estimator detailed in Algorithm 3 uses the algorithm for locally computing the labels of query points for a nearly optimal function from the previous section. In particular, it samples a few random query points and uses them to compute the average empirical error. The final query complexity of our procedure is then obtained via standard concentration arguments, relating the empirical error to the true expected error. We formalize this guarantee in Theorem 2 and defer its proof to Appendix A.

Algorithm 3 Error$(L, \mathcal{D}, \epsilon)$

```
Let $S, \mathcal{P}$ =Preprocess$(L, \mathcal{D}_x, \epsilon)$
Sample a set $\{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$ labeled examples from distribution $\mathcal{D}$ where $N = O(\frac{1}{\epsilon^2})$
**Output** $\widehat{\mathrm{err}}_{\mathcal{D}}(\mathcal{F}_L) = \frac{1}{N}\sum_{i=1}^{N} |\mathrm{Query}(x_i, S, \mathcal{P}) - y_i|$
```

Theorem 2 *For any distribution $\mathcal{D}$ over $[0,1] \times [0,1]$, Lipschitz constant $L > 0$ and parameter $\epsilon \in [0,1]$, Algorithm 3 uses $O(\frac{1}{\epsilon^6}\log(\frac{1}{\epsilon}))$ active label queries on $O(\frac{L}{\epsilon^4}\log(\frac{1}{\epsilon}))$ unlabeled samples from distribution $\mathcal{D}_x$ and produces an output $\widehat{err}_{\mathcal{D}}(\mathcal{F}_L)$ satisfying*

$$|\widehat{err}_{\mathcal{D}}(\mathcal{F}_L) - err_{\mathcal{D}}(\mathcal{F}_L)| \leq \epsilon$$

with probability at least $\frac{1}{2}$.

5. Nadaraya-Watson Estimator

In this section, we consider the related problem of approximating the minimum error that can be achieved by the Nadaraya-Watson estimator under a linear diagonal transformation with eigenvalues coming from a small range. In this setting, there exists a distribution $\mathcal{D}$ over the domain $\mathbb{R}^d$. Each data point $x \in \mathbb{R}^d$ has a true label $f(x) \in \{0,1\}$. The Nadaraya-Watson prediction algorithm when given a dataset $S = \{x_1, x_2, \cdots, x_N\}$ of unlabeled samples sampled from distribution $\mathcal{D}$ and a query point x outputs the prediction

$$\tilde{f}_{S,K_A}(x) = \frac{\sum_{i=1}^{N} K_A(x_i, x) f(x_i)}{\sum_{i=1}^{N} K_A(x_i, x)} = \sum_{i=1}^{N} p_{S,A}(x_i, x) f(x_i)$$

where $K_A(x, y) = 1/(1+||A(x-y)||_2^2)$ is the kernel function for matrices $A \in \mathbb{R}^{d\times d}$. The loss of the data point $(x, f(x))$ with respect to the unlabeled samples S and the kernel K_A is

$$l_{S,K_A}(x) = |f(x) - \tilde{f}_{S,K_A}(x)|$$

The total loss of the prediction function $\tilde{f}_{S,K_A}$ with respect to distribution $\mathcal{D}$ is

$$L_{S,K_A} = E_{x\sim\mathcal{D}}|f(x) - \tilde{f}_{S,K_A}(x)|$$

Now, let us say we are interested in computing the prediction loss with the best diagonal linear transformation A for the data with eigenvalues bounded between constants 1 and 2 that is a matrix $A \in \mathcal{A}$ where $\mathcal{A} = \{A \in \mathbb{R}^{d\times d} | A_{i,j} = 0\ \forall i \neq j \text{ and } 1 \leq A_{i,i} \leq 2\}$ that is

$$L_S = \min_{A\in\mathcal{A}} L_{S,K_A} = \min_{A\in\mathcal{A}} \mathop{\mathbb{E}}_{x\sim\mathcal{D}} \sum_i p_{S,A}(x_i, x)|f(x_i) - f(x)|$$

In this section, we show (Theorem 4) that it is possible to estimate the loss L_S with an additive error of ϵ with labeled sample complexity $\tilde{O}(d/\epsilon^2)$ and running time $\tilde{O}(d^2/\epsilon^{d+4} + dN/\epsilon^2)$. We use $[n]$ to denote $\{1, 2, \cdots, n\}$ for a positive integer n and $\tilde{O}$ notation to hide polylogarithmic factors in the input parameters and error rates.

First, we discuss a theorem from Backurs et al. (2018) which will be crucial in the proof of Theorem 4. Theorem 19, formally stated in the appendix, states that for certain nice kernels $K(x, y)$, it is possible to efficiently estimate $N^{-1}\sum_{i=1}^{N} K(x, x_i)$ with a multiplicative error of ϵ for any query x. As a direct corollary of Theorem 19, we obtain that it is possible to efficiently estimate the probabilities $p_{S,A}(q, x_i)$, for all data points x_i in S, queries $q \in \mathbb{R}^d$ and matrices A in $\mathcal{A}$. Let us define $\hat{S}_{S,A}(q)$ to be the estimator for $\sum_{x_i\in S} K(q, x_i)$ as per Theorem 19.

Corollary 3 *There exists a data structure that given a data set $S \subset \mathbb{R}^d$ with $|S| = N$, using $O(\frac{Nd}{\epsilon^2}\log(\frac{N}{\delta}))$ space and preprocessing time, for any $A \in \mathcal{A}$ and a query $q \in \mathbb{R}^d$ and data point $x \in \mathbb{R}^d$, estimates $p_{S,A}(q, x) = K_A(q,x)/\sum_{y\in S} K_A(q,y)$ by using the estimator $\hat{p}_{S,A}(q, x) = K_A(q,x)/\hat{S}_{S,A}(q)$ with accuracy $(1 \pm \epsilon)$ in time $O(\log(\frac{N}{\delta})\frac{d}{\epsilon^2})$ with probability at least $1 - 1/poly(N) - \delta$.*

We state the algorithm NW Error (Algorithm 4) for computing the minimum prediction error of the algorithm with respect to underlying distribution $\mathcal{D}$, the set of unlabeled training data S, the set of matrices $\mathcal{A}$, the error parameter ϵ and the failure probability δ and discuss the idea behind the algorithm in the next few paragraphs.

Naive Algorithm. The naive algorithm would take $O(1/\epsilon^2)$ labeled data points sampled from distribution $\mathcal{D}$ for each of the matrices $A \in \mathcal{A}_\epsilon$, an ϵ-cover of the set $\mathcal{A}$ (Lemma 20) of size $O(1/\epsilon^d)$ and compute the exact loss using the N data points in the training set. Hence, the number of labels required in this algorithm is $N + O(1/\epsilon^{d+2})$.

Dependence on N of label query complexity. Using our algorithm, we achieve labeled sample complexity of $\tilde{O}(d/\epsilon^2)$, independent of N and depending only polynomially on d. For getting rid of the dependence on N, the idea is to first sample $O(1/\epsilon^2)$ samples from distribution $\mathcal{D}$ and then for each sample, sample a training data point from S with probability proportional to $p_{S,A}$ for the matrix A. This gets rid of the dependence on N. However, we still have to repeat this procedure separately for every matrix $A \in \mathcal{A}$ which leads to a requirement of $O(1/\epsilon^{d+2})$ labels.

Algorithm 4 NW Error$(S, \mathcal{A}, \mathcal{D}, \epsilon, \delta)$

```
Let $\mathcal{A}_\epsilon = \{A \in \mathbb{R}^{d\times d} \mid A_{i,j} = 0 \; \forall\, i \neq j$ and $A_{i,i} \in \{1, 1+\epsilon, (1+\epsilon)^2, \cdots, 2\} \; \forall i \in [d]\}$.
Sample $M = O\left(\frac{1}{\epsilon^2}\log\left(\frac{|\mathcal{A}_\epsilon|}{\delta}\right)\right)$ labeled examples $\{(z_i, f(z_i))\}_{i=1}^M$ with each $(z_i, f(z_i)) \sim \mathcal{D}$.
for $i = 1$ to $M$ do
    Sample a $\tilde{z}_i$ with probability proportional to $p_{S,I}(z_i, \tilde{z}_i)$.
end for
for $A \in \mathcal{A}_\epsilon$ do
    for $i = 1$ to $M$ do
        Compute $\hat{p}_{S,A}(z_i, \tilde{z}_i) = \frac{K_A(z_i, \tilde{z}_i)}{\hat{S}_{S,A}(z_i)}$.
    end for
    Compute $\hat{L}_{S,K_A} = \frac{1}{M}\sum_{i=1}^M |f(z_i) - f(\tilde{z}_i)| \frac{\hat{p}_{S,A}(z_i, \tilde{z}_i)}{p_{S,I}(z_i, \tilde{z}_i)}$.
end for
Output $\hat{L}_S = \min_{A \in \mathcal{A}_\epsilon} \hat{L}_{S,K_A}$.
```

Dependence on d of label query complexity. To eliminate the exponential dependence on d, we show that for matrices in $\mathcal{A}$, we can use importance sampling and the samples generated for the identity matrix I suffice to estimate the loss for all matrices $A \in \mathcal{A}$ with appropriate scaling factors $p_{S,A}/p_{S,I}$. This is because the eigenvalues of all the matrices are bounded between constants and hence, the sampling probabilities $p_{S,A}$ are similar up to a multiplicative factor (Lemma 21). This leads to our desired labeled sample complexity of $\tilde{O}(d/\epsilon^2)$. The factor d comes in because of using a union bound over all the exponential number of matrices in $\mathcal{A}_\epsilon$.

Running time. However, using this approach directly, we obtain a running time of $\tilde{O}(Nd/\epsilon^{d+2})$ because for each matrix A, for each sample, we have to compute $p_{S,A}/p_{S,I}$ which requires going over all the data points in the set S. To achieve better running times, we use the faster kernel density estimation algorithm (Backurs et al., 2018) to compute approximate probabilities efficiently (Corollary 3) and obtain a running time of $\tilde{O}(d^2/\epsilon^{d+4} + Nd/\epsilon^2)$ separating the multiplicative dependence of N and $1/\epsilon^d$. We state the formal guarantees in Theorem 4 and its proof in Appendix C.

Theorem 4 *For a d-dimensional unlabeled pointset S with $|S| = N$, Algorithm 4 queries $O(\frac{1}{\epsilon^2}(d\log(\frac{1}{\epsilon}) + \log(\frac{1}{\delta})))$ labels from S and outputs $\hat{L}_S$ such that*

$$|\hat{L}_S - \min_{A \in \mathcal{A}} \mathop{\mathbb{E}}_{x \sim \mathcal{D}} \sum_i p_{S,A}(x_i, x)|f(x_i) - f(x)|| \leq \epsilon$$

with a failure probability of at most $\delta + \frac{d}{\epsilon^{d+2} poly(N)} \log(\frac{1}{\epsilon\delta})$ and runs in time $\tilde{O}(\frac{d^2}{\epsilon^{d+4}} + \frac{dN}{\epsilon^2})$.

6. Conclusion

We gave an algorithm to approximate the optimal prediction error for the class of bounded L-Lipschitz functions with independent of L label queries. We also established that for any given query point, we can estimate the value of a nearly optimal function at the query point locally with label queries, independent of L. It would be interesting to extend these notions of error prediction and local prediction to other function classes. Finally, we also gave an algorithm to approximate the minimum error of the Nadaraya-Watson prediction rule under a linear diagonal transformation with eigenvalues in a small range which is both sample and time efficient.

Acknowledgments

We thank the anonymous reviewers who helped improve the readability and presentation of this draft by providing many helpful comments. This work was done in part while Neha Gupta was visiting TTIC.

References

Noga Alon, Ronitt Rubinfeld, Shai Vardi, and Ning Xie. Space-efficient local computation algorithms. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 1132–1139. Society for Industrial and Applied Mathematics, 2012.

Alexandr Andoni, Robert Krauthgamer, and Yosef Pogrow. On solving linear systems in sublinear time. In *10th Innovations in Theoretical Computer Science Conference (ITCS 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

Martin Anthony and Peter L Bartlett. *Neural network learning: Theoretical foundations*. cambridge university press, 2009.

Arturs Backurs, Moses Charikar, Piotr Indyk, and Paris Siminelakis. Efficient density evaluation for smooth kernels. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 615–626. IEEE, 2018.

Maria-Florina Balcan, Eric Blais, Avrim Blum, and Liu Yang. Active property testing. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 21–30. IEEE, 2012.

Mihir Bellare, Don Coppersmith, JOHAN Hastad, Marcos Kiwi, and Madhu Sudan. Linearity testing in characteristic two. *IEEE Transactions on Information Theory*, 42(6):1781–1795, 1996.

Eli Ben-Sasson, Madhu Sudan, Salil Vadhan, and Avi Wigderson. Randomness-efficient low degree tests and short pcps via epsilon-biased sets. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 612–621. ACM, 2003.

Piotr Berman, Sofya Raskhodnikova, and Grigory Yaroslavtsev. Lp-testing. In *STOC*, pages 164–173, 2014.

Avrim Blum and Lunjia Hu. Active tolerant testing. In *Conference On Learning Theory*, pages 474–497, 2018.

Léon Bottou and Vladimir Vapnik. Local learning algorithms. *Neural computation*, 4(6):888–900, 1992.

Deeparnab Chakrabarty and Comandur Seshadhri. Optimal bounds for monotonicity and lipschitz testing over hypercubes and hypergrids. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 419–428. ACM, 2013.

Deeparnab Chakrabarty and Comandur Seshadhri. An o(n) monotonicity tester for boolean functions over the hypercube. *SIAM Journal on Computing*, 45(2):461–472, 2016.

Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

Xi Chen, Rocco A Servedio, and Li-Yang Tan. New algorithms and lower bounds for monotonicity testing. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 286–295. IEEE, 2014.

Ilias Diakonikolas, Homin K Lee, Kevin Matulef, Krzysztof Onak, Ronitt Rubinfeld, Rocco A Servedio, and Andrew Wan. Testing for concise representations. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 549–558. IEEE, 2007.

Uriel Feige, Yishay Mansour, and Robert Schapire. Learning and inference in the presence of corrupted inputs. In *Conference on Learning Theory*, pages 637–657, 2015.

Oded Goldreich, Shari Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM (JACM)*, 45(4):653–750, 1998a.

Oded Goldreich, S Goldwassert, Eric Lehman, and Dana Ron. Testing monotonicity. In *Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No. 98CB36280)*, pages 426–435. IEEE, 1998b.

Lee-Ad Gottlieb, Aryeh Kontorovich, and Robert Krauthgamer. Efficient regression in metric spaces via approximate lipschitz extension. *IEEE Transactions on Information Theory*, 63(8):4838–4849, 2017.

Christoph Grunau, Slobodan Mitrović, Ronitt Rubinfeld, and Ali Vakilian. Improved local computation algorithm for set cover via sparsification. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2993–3011. SIAM, 2020.

Madhav Jha and Sofya Raskhodnikova. Testing and reconstruction of lipschitz functions with applications to data privacy. *SIAM Journal on Computing*, 42(2):700–731, 2013.

Michael Kearns and Dana Ron. Testing problems with sublearning sample complexity. *Journal of Computer and System Sciences*, 61(3):428–456, 2000.

Subhash Khot, Dor Minzer, and Muli Safra. On monotonicity testing and boolean isoperimetric-type theorems. *SIAM Journal on Computing*, 47(6):2238–2276, 2018.

Weihao Kong and Gregory Valiant. Estimating learnability in the sublinear data regime. In *Advances in Neural Information Processing Systems*, pages 5455–5464, 2018.

Reut Levi, Dana Ron, and Ronitt Rubinfeld. Local algorithms for sparse spanning graphs. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, page 826, 2014.

Yishay Mansour and Shai Vardi. A local computation approximation scheme to maximum matching. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 260–273. Springer, 2013.

Yishay Mansour, Aviad Rubinstein, Shai Vardi, and Ning Xie. Converting online algorithms to local computation algorithms. In *International Colloquium on Automata, Languages, and Programming*, pages 653–664. Springer, 2012.

Yishay Mansour, Aviad Rubinstein, and Moshe Tennenholtz. Robust probabilistic inference. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 449–460. SIAM, 2014.

Edward James McShane. Extension of range of functions. *Bulletin of the American Mathematical Society*, 40(12):837–842, 1934.

Michal Parnas and Dana Ron. Approximating the minimum vertex cover in sublinear time and a connection to distributed algorithms. *Theoretical Computer Science*, 381(1-3):183–196, 2007.

Michal Parnas, Dana Ron, and Alex Samorodnitsky. Testing basic boolean formulae. *SIAM Journal on Discrete Mathematics*, 16(1):20–46, 2002.

Michal Parnas, Dana Ron, and Ronitt Rubinfeld. Tolerant property testing and distance approximation. *Journal of Computer and System Sciences*, 72(6):1012–1042, 2006.

Merav Parter, Ronitt Rubinfeld, Ali Vakilian, and Anak Yodpinyanee. Local computation algorithms for spanners. *10th Innovations in Theoretical Computer Science*, 2019.

Ronitt Rubinfeld, Gil Tamir, Shai Vardi, and Ning Xie. Fast local computation algorithms. *arXiv preprint arXiv:1104.1377*, 2011.

Vladimir Vapnik. Principles of risk minimization for learning theory. In *Advances in neural information processing systems*, pages 831–838, 1992.

Vladimir Vapnik. Statistical learning theory, 1998.

Appendix A. One Dimensional Case

We first re-state the Theorem 2 which gives sample complexity guarantees for error estimation for the class of one-dimensional Lipschitz functions and the proof.

Theorem 2 *For any distribution $\mathcal{D}$ over $[0,1] \times [0,1]$, Lipschitz constant $L > 0$ and parameter $\epsilon \in [0,1]$, Algorithm 3 uses $O(\frac{1}{\epsilon^6}\log(\frac{1}{\epsilon}))$ active label queries on $O(\frac{L}{\epsilon^4}\log(\frac{1}{\epsilon}))$ unlabeled samples from distribution $\mathcal{D}_x$ and produces an output $\widehat{err_{\mathcal{D}}}(\mathcal{F}_L)$ satisfying*

$$|\widehat{err_{\mathcal{D}}}(\mathcal{F}_L) - err_{\mathcal{D}}(\mathcal{F}_L)| \leq \epsilon$$

with probability at least $\frac{1}{2}$.

Proof By Theorem 1, we know that $\text{Query}(x, S, \mathcal{P}) = \tilde{f}(x) \; \forall x \in [0,1]$ and the error of $\tilde{f}$ additively approximates the error of function $f^*_{\mathcal{D}}$, that is,

$$\Delta_{\mathcal{D}}(\tilde{f}, \mathcal{F}_L) = \text{err}_{\mathcal{D}}(\tilde{f}) - \text{err}_{\mathcal{D}}(\mathcal{F}_L) \leq \epsilon$$

with probability greater than $1/2$. Thus, with probability greater than $1/2$, we get

$$\begin{aligned} |\widehat{\text{err}_{\mathcal{D}}}(\mathcal{F}_L) - \text{err}_{\mathcal{D}}(\tilde{f})| &= |\frac{1}{N}\sum_{i=1}^{N} |\text{Query}(x_i, S, \mathcal{P}) - y_i| - \text{err}_{\mathcal{D}}(\tilde{f})| \\ &= |\frac{1}{N}\sum_{i=1}^{N} |\tilde{f}(x_i) - y_i| - \text{err}_{\mathcal{D}}(\tilde{f})| \leq \epsilon \end{aligned}$$

The last inequality follows by standard concentration arguments since $N \geq O(1/\epsilon^2)$. The theorem statement follows by using triangle inequality.

By Theorem 1, the number of unlabeled samples is $O((L/\epsilon^4)\log(1/\epsilon))$ and the number of label queries is $O((1/\epsilon^2) \cdot ((1/\epsilon^4)\log(1/\epsilon))) = O((1/\epsilon^6)\log(1/\epsilon))$. ■

Now, we will state the lemmas involved in the proof of Theorem 1 with their proofs. The following lemma proves that with enough unlabeled samples, a large fraction of long intervals have enough unlabeled samples in them which is eventually used to argue that they will be sufficient to learn a function which is approximately close to the optimal Lipschitz function over that interval.

Lemma 5 *For any distribution $\mathcal{D}_x$, consider a set $S = \{x_1, x_2, \cdots, x_M\}$ of unlabeled samples where each sample $x_i \overset{i.i.d.}{\sim} \mathcal{D}_x$. Let $\mathcal{G}$ be the set of long intervals $\{I_i\}$ each of which satisfies $p_i = \Pr_{x\sim\mathcal{D}_x}(x \in I_i) \geq \frac{1}{L}$. Let E_i denote the event that $\sum_{x_j \in S} \mathbb{I}[x_j \in I_i] < \frac{1}{2\epsilon^4}\log(\frac{1}{\epsilon})$. Then, we have*

$$\sum_{I_i \in \mathcal{G}} p_i \mathbb{I}[E_i] \leq \frac{\epsilon}{\delta}$$

with failure probability atmost δ for $M = \Omega(\frac{L}{\epsilon^4}\log(\frac{1}{\epsilon}))$.

Proof For any interval $I \in \mathcal{G}$, we have that $\mathbb{E}[\sum_{x_j \in S} \mathbb{I}[x_j \in I]] \geq \frac{1}{\epsilon^4}\log(\frac{1}{\epsilon})$. Using Hoeffding inequality, we can get that $\Pr(E_i) \leq \epsilon$ for all intervals $I_i \in \mathcal{G}$. Calculating expectation of the desired quantity, we get

$$\mathbb{E}[\sum_{I_i \in \mathcal{G}} p_i \mathbb{I}[E_i]] = \sum_{I_i \in \mathcal{G}} p_i \Pr[E_i] \leq \epsilon \sum_{I_i \in \mathcal{G}} p_i \leq \epsilon$$

We get the desired result using Markov's inequality. ■

The following lemma states that the probability of short intervals p_{sh} is small with high probability. Consider the case of uniform distributions. In this case, since the short intervals cover only ϵ fraction of the $[0,1]$ length, their probability p_{sh} is upper bounded by ϵ. The case for arbitrary distributions holds because the intervals are chosen randomly.

Lemma 6 *When we divide the $[0,1]$ domain into alternating intervals of length $\frac{1}{L\epsilon}$ and $\frac{1}{L}$ with a random offset at $\{0,1,2,\cdots,\frac{1}{\epsilon}\}\frac{1}{L}$ as in the preprocessing step, then*

$$p_{\mathsf{sh}} = \sum_{I_i \in \mathcal{P}_{\mathsf{sh}}} p_i \leq \frac{\epsilon}{\delta}$$

with failure probability atmost δ.

Proof Now, if we consider the division of $[0,1]$ into alternating intervals of length $1/(L\epsilon)$ and $1/L$ with the offset chosen uniformly randomly from $\{0,1,2,\cdots,1/\epsilon\}(1/L)$, then the intervals of length $1/L$ combined are disjoint in each of these divisions and together cover the entire $[0,1]$ length. Hence, there are at most δ fraction out of the total $1/\epsilon$ cases where the short intervals have probability greater than ϵ/δ. Hence, with probability $1-\delta$, the short intervals have probability upper bounded by ϵ/δ. ■

Lemma 7 *Let $I_i \in \mathcal{P}_{\mathsf{lg},1}$ be any long interval of subtype 1. For the event $F_i = \{\Delta_{\mathcal{D}_i}(\hat{f}_{S\cap I_i}, f^*_{\mathcal{D}_i}) > \epsilon\}$, we have*

$$\Pr(F_i) \leq \epsilon$$

Proof We know that the covering number for the class of one dimensional L-Lipschitz functions supported on the interval $[0,l]$ is $O(Ll/\epsilon)$ (Lemma 12). For, Lipschitz functions supported on a long interval of length $l = 1/(L\epsilon)$, we get this complexity as $O(1/\epsilon^2)$. We know by standard results in uniform convergence, that the number of samples required for uniform convergence up to an error of ϵ and failure probability δ for all functions in a class $\mathcal{F}_L$ is $\Omega(((\text{Covering number of } \mathcal{F}_L)/\epsilon^2)\log(\frac{1}{\delta}))$ (Lemma 10) and hence, for long intervals we get this estimate as $\Omega(1/\epsilon^4 \log(1/\delta))$. Given that there are $\Omega((1/\epsilon^4)\log(1/\epsilon))$ samples in these intervals by design, the function learned using these samples is only additively worse than the function with minimum error using standard learning theory arguments with failure probability at most ϵ. Thus, we get that

$$\Delta_{\mathcal{D}_i}(f_{S\cap I_i}, f^*_{\mathcal{D}_i}) \leq \epsilon \quad \forall I_i \in \mathcal{P}_{\mathsf{lg},1}$$

with failure probability at most ϵ. ■

Now, we state the definitions of covering number of a metric space and the uniform covering number of a hypothesis class. These definitions are used in Lemma 10 to argue how fast the empirical error converges to the expected error for a given hypothesis class.

Definition 8 *$N(\epsilon, A, \rho)$ is the covering number of the metric space A with respect to distance measure ρ at scale ϵ and is defined as*

$$N(\epsilon, A, \rho) = \min\{|C| \mid C \textit{ is an } \epsilon\textit{-cover of } A \textit{ wrt } \rho \ (\forall x \in A, \exists c \in C \textit{ st } \rho(c,x) \leq \epsilon)\}.$$

Definition 9 *$N_p(\epsilon, F, m)$ for $p \in \{1, 2, \infty\}$ is the uniform covering number of the hypothesis class F at scale ϵ with respect to distance measure d_p where $d_p(x, y) = ||x - y||_p$ and is defined as*

$$N_p(\epsilon, F, m) := \max_{x \in X^m} N(\epsilon, \{f(x_1), f(x_2), \cdots, f(x_m)\}, d_p)$$

where $f \in F$ and $N(\epsilon, A, \rho)$ is as defined in definition 8.

The following lemma uses the well known generalization theory to argue how fast the empirical error uniformly converges to the expected error for the class of d-dimensional Lipschitz functions.

Lemma 10 *Let $S = \{(x_i, y_i)\}_{i=1}^{M}$ be a set of $M > \frac{1}{\epsilon^2}(\Omega(\frac{Ll}{\epsilon}))^d \log(\frac{1}{\delta})$ data points sampled uniformly randomly from distribution $\mathcal{D}$. Then,*

$$err_{\mathcal{D}}(\hat{f}_S) - err_{\mathcal{D}}(\mathcal{F}_L) \leq \epsilon$$

with probability at least $1 - \delta$.

Proof A standard result for uniform convergence for general loss functions (For example, Theorem 21.1 in Anthony and Bartlett (2009)) states that

$$\Pr_{S \sim \mathcal{D}^n}[\sup_{f \in F} |er^l_{\mathcal{D}}[f] - er^l_S[f]|| > \epsilon] \leq 4N_1(\frac{\epsilon}{8}, l_F, 2m)e^{\frac{-m\epsilon^2}{32}}$$

where l is the loss function bounded between $[0, 1]$ and F is a class of functions mapping into $[0, 1]$.

We will prove that $N_1(\frac{\epsilon}{8}, l_F, 2m) \leq (O(\frac{Ll}{\epsilon}))^d$ which will complete the proof of the lemma by using triangle inequality and uniform convergence for $f^*_{\mathcal{D}}$ and $\hat{f}_S$.

$$\log(N_1(\frac{\epsilon}{8}, l_F, 2m)) \overset{\text{(i)}}{\leq} \log(N_1(\frac{\epsilon}{8}, F, 2m)) \overset{\text{(ii)}}{\leq} \log(N_\infty(\frac{\epsilon}{8}, F, 2m)) \overset{\text{(iii)}}{\leq} (O(\frac{Ll}{\epsilon}))^d$$

(i) follows because $|l(f(x_i), y_i) - l(f'(x_i), y_i)| \leq |f(x_i) - f'(x_i)|$ for all $f, f' \in F$. (ii) follows from Lemma 10.5 in Anthony and Bartlett (2009) and (iii) follows from Lemma 12. ■

Now, we state the definition of Lipschitz extension in Theorem 11 and the result which states that for any metric space, if we have a L-Lipschitz function on a subset of the metric space, then it is possible to extend the function to the entire space with respect to the metric which preserves the values of the function at the points originally in the domain and is now L-Lipschitz on the entire domain. This will be used in the computing covering number bounds for the class of Lipschitz functions (Lemma 12).

Theorem 11 *[Theorem 1 from McShane (1934)] For a L-Lipschitz function $f : E \to \mathbb{R}$ defined on a subset E of the metric space S, f can be extended to S such that its values on the subset E is preserved and it satisfies the L Lipschitz property over the entire domain S with respect to the same metric. Such an extension is called Lipschitz extension of the function f.*

Now, we will state the well known covering number bounds for the class of high dimensional Lipschitz functions. Note that we have stated this here just for completeness and the proof essentially follows the proof from Gottlieb et al. (2017).

Lemma 12 *For the class of high dimensional Lipschitz functions* $\mathcal{F}_L : [0,l]^d \to [0,1]$ *where* $f \in \mathcal{F}_L$ *satisfies* $|f(x) - f(y)| \leq L||x-y||_\infty$ $\forall x, y \in [0,l]^d$*, we have* $\log(N_\infty(\epsilon, F, m)) = (O(\frac{Ll}{\epsilon}))^d$.

Proof Let us consider a discretization of the domain $P = [0,l]^d$ where we divide each coordinate of the domain into intervals of length $\frac{\epsilon}{3L}$. Let us consider a set $\mathcal{F}_L^\epsilon$ of all those Lipschitz functions which are the Lipschitz extensions of the Lipschitz functions which have output values amongst $R = \{0, \frac{\epsilon}{3}, \frac{2\epsilon}{3}, \cdots, 1\}$ at the discretized points of the domain P (note that it is always possible to form a Lipschitz extension of a Lipschitz function over metric space by McShane (1934), also mentioned in Theorem 11 above). So, we have that $|\mathcal{F}_L^\epsilon| \leq (\frac{3}{\epsilon})^{(\frac{3Ll}{\epsilon})^d}$.

Now, we will show that $\mathcal{F}_L^\epsilon$ forms a valid covering of the function class $\mathcal{F}_L$, and hence we get $\log(N_\infty(\epsilon, \mathcal{F}_L, m)) = (O(\frac{Ll}{\epsilon}))^d$. Now, let us show that for any $f \in \mathcal{F}_L$, there exists a function $\tilde{f} \in \mathcal{F}_L^\epsilon$ such that $\sup_x |f(x) - \tilde{f}(x)| \leq \epsilon$. Consider a function $\hat{f}$ such that $\hat{f}(x) = \operatorname{argmin}_{y \in R} |y - f(x)|$ at the discretization of the domain P and let $\tilde{f}$ be its Lipschitz extension. First, we will argue that $\hat{f}(x)$ is L Lipschitz and since $\tilde{f}$ is a Lipschitz extension of $\hat{f}$, $\tilde{f}$ is also L Lipschitz and by construction belongs to F_L^ϵ. Now, we have $|\hat{f}(x) - \hat{f}(y)| \leq |f(x) - f(y)| + \frac{2\epsilon}{3} \leq \epsilon$ since f is L-Lipschitz and $||x - y||_\infty \leq \frac{\epsilon}{3L}$ by construction and hence, we see that $\hat{f}$ is L-Lipschitz. Now, we will show that $\sup_x |f(x) - \tilde{f}(x)| \leq \epsilon$. Consider any point $x \in [0,l]^d$. Now, we know that there exists a $\tilde{x} \in P$ i.e. in the discretization of the domain such that $||x - \tilde{x}||_\infty \leq \frac{\epsilon}{3L}$. Hence, we get that $|\tilde{f}(x) - f(x)| \leq |\tilde{f}(x) - \tilde{f}(\tilde{x})| + |\tilde{f}(\tilde{x}) - f(\tilde{x})| + |f(\tilde{x}) - f(x)| \leq \frac{\epsilon}{3} + \frac{\epsilon}{3} + \frac{\epsilon}{3} \leq \epsilon$ since the functions $\tilde{f}$ and f are both L-Lipschitz and $|\tilde{f}(\tilde{x}) - f(\tilde{x})| = |\hat{f}(\tilde{x}) - f(\tilde{x})| \leq \frac{\epsilon}{3}$. ■

Appendix B. High Dimensional Case

B.1. Problem Setup

We recall the setting from the one dimensional case. For any fixed $L > 0$, let $\mathcal{F}_L$ be the class of d dimensional functions supported on the domain $[0,1]^d$ with Lipschitz constant at most L, that is,

$$\mathcal{F}_L = \{f : [0,1]^d \mapsto [0,1],\ |f(x) - f(y)| \leq L\|x - y\|_\infty\ \forall x, y \in [0,1]^d\}. \tag{8}$$

Let ϵ be the error parameter. We will think of the dimension d as constant with respect to L and ϵ. Like the one-dimensional case, our algorithm for local predictions first involves a preprocessing step (Algorithm 5) which takes as input the Lipschitz constant L, sampling access to distribution $\mathcal{D}_x$ and the error parameter ϵ and returns a partition $\mathcal{P}$ where the partition along dimension j is $\mathcal{P}^j = \{[b_0^j, b_1^j], [b_1^j, b_2^j], \ldots, \}$ and a set S of unlabeled samples. The partition $\mathcal{P}$ consists of alternating intervals[3] of length $2/L$ and $d/(L\epsilon)$ along each dimension. Let us divide these intervals along each dimension j further into the two sets

$$\mathcal{P}^j_{\mathsf{lg}} := \{[b_0, b_1], [b_2, b_3], \ldots, \} \quad \text{(long intervals for dimension } j \text{ of length } d/(L\epsilon)),$$
$$\mathcal{P}^j_{\mathsf{sh}} := \{[b_1, b_2], [b_3, b_4], \ldots, \} \quad \text{(short intervals for dimension } j \text{ of length } 2/L).$$

A data point x which belongs to a short interval in $\mathcal{P}^j$ along at least one of the dimensions j is said to belong to a set of short intervals $\mathcal{P}_{\mathsf{sh}}$ and otherwise, is said to belong to a set of long intervals $\mathcal{P}_{\mathsf{lg}}$.

[3] Note that long intervals at the boundary could be shorter, but those can be handled similarly.

Algorithm 5 Preprocess($L, \mathcal{D}_x, \epsilon$)

```
Sample a uniformly random offset $b_1^j$ from $\{0, 1, 2, \cdots, \frac{d}{2\epsilon}\}\frac{2}{L}$ for each dimension $j \in [d]$.
Divide the $[0, 1]$ interval along each dimension $j$ into alternating intervals of length $\frac{d}{L\epsilon}$ and $\frac{2}{L}$ with boundary at $b_1^j$ and let $\mathcal{P}$ be the resulting partition, that is, $\mathcal{P}^j = \{[b_0^j = 0, b_1^j], [b_1^j, b_2^j], \ldots, ], \cdots, 1\}$ where $b_2^j = b_1^j + \frac{2}{L}, b_3^j = b_2^j + \frac{d}{L\epsilon}, \ldots,$.
Sample a set $S = \{x_i\}_{i=1}^M$ of $M = O((\frac{L}{\epsilon})^d \frac{1}{\epsilon^3} \log(\frac{1}{\epsilon}))$ unlabeled examples from distribution $\mathcal{D}_x$.
Output $S, \mathcal{P}$.
```

We give the definition of the extension interval for a long interval in $\mathcal{P}_{\mathsf{lg}}$. It consists of a long interval and a cuboidal shell of thickness $\frac{1}{L}$ around it in each dimension and cut off at the boundary.

Definition 13 *For any long interval I_J where $I_J^i = [b^i_{J^i-1}, b^i_{J^i}]$, the extension interval is defined to be $\hat{I}_J$ where $\hat{I}_J^i = [\max(0, b^i_{J^i-1} - \frac{1}{L}), \min(b^i_{J^i} + \frac{1}{L}, 1)]\ \forall i \in [d]$ such that $I_J \subset \hat{I}_J$. In other words, interval $\hat{I}_J$ consists of the interval I_J and a cuboidal shell of thickness $\frac{1}{L}$ around it on both sides in each dimension unless it does not extend beyond $[0, 1]$ in each dimension.*

The Query algorithm (Algorithm 6) for test point x^* takes as input the set S of unlabeled examples and the partition $\mathcal{P}$ returned by the Preprocess algorithm. Note that all subsequent queries use the same partition $\mathcal{P}$ and the same set of unlabeled examples S. The algorithm uses different learning strategies depending on whether x^* belongs to one of the long intervals in $\mathcal{P}_{\mathsf{lg}}$ or short intervals in $\mathcal{P}_{\mathsf{sh}}$.

For the long intervals, it outputs the prediction corresponding to the empirical risk minimizer (ERM) function restricted to that interval. The middle function values of the short intervals are constrained to be 1 which makes the overall function Lipschitz. If a query point x lies in a short interval which is the Lipschitz extension of a given long interval, the function learned is the Lipschitz extension of function over the long interval. This ensures that the overall function is Lipschitz. We bound the expected prediction error of this scheme with respect to class $\mathcal{F}_L$ by separately bounding this error for long and short intervals. For the long intervals, we prove that the ERM has low error by ensuring that each interval contains enough unlabeled samples. On the other hand, we show that the short intervals do not contribute much to the error because of their low probability under the distribution $\mathcal{D}$.

Algorithm 6 Query($x, S, \mathcal{P} = [\{[b_0^j, b_1^j], [b_1^j, b_2^j], \cdots\}]_{j=1}^d$)

```
if query $x \in I_J$ where $I_J^j = [b^j_{J^j-1}, b^j_{J^j}]$ when $I_J \in \mathcal{P}_{\mathsf{lg}}$ then
    Query labels for $x \in S \cap I_J$
    Output $\hat{f}_{S \cap I_J}(x)$
else if query $x \in \hat{I}_J$ where $\hat{I}_J$ is the extension of a long interval $I_J$ where $I_J^j = [b^j_{J^j-1}, b^j_{J^j}]$ is a long interval then
    Query labels for $x \in S \cap I_J$
    Output: $f(x)$ where $f(x)$ is the Lipschitz extension of $f_J(x)$ to the extension set $\hat{I}_J$ with constraints $f(x) = 1 \; \forall x \in \hat{I}_J$ satisfying $x_i \in \left\{\frac{b^i_{J^i}+b^i_{J^i+1}}{2}, \frac{b^i_{J^i-2}+b^i_{J^i-1}}{2}\right\}$ for at least one $i \in [d]$.
else
    Output: 1
end if
```

Next, we state the number of label queries needed to making local predictions corresponding to the Query algorithm (Algorithm 6).

Theorem 14 *For any distribution $\mathcal{D}$ over $[0,1]^d \times [0,1]$, Lipschitz constant $L > 0$ and error parameter $\epsilon \in [0,1]$, let (S, P) be the output of (randomized) Algorithm 1 where S is the set of unlabeled samples of size $(O(\frac{L}{\epsilon}))^d \frac{1}{\epsilon^3} \log(\frac{1}{\epsilon})$ and P is a partition of the domain $[0,1]^d$. Then, there exists a function $\tilde{f} \in \mathcal{F}_L$, such that for all $x \in [0,1]$, Algorithm 6 queries $\frac{1}{\epsilon^2}(O(\frac{d}{\epsilon^2}))^d \log(\frac{1}{\epsilon})$ labels from the set S and outputs Query(x, S, P) satisfying*

$$\textit{Query}(x, S, P) = \tilde{f}(x) \;, \tag{9}$$

and the function $\tilde{f}$ is ϵ-optimal, that is, $\Delta_{\mathcal{D}}(\tilde{f}, \mathcal{F}_L) \leq \epsilon$ with probability greater than $\frac{1}{2}$.

Proof We begin by defining some notation. Let $S = \{x_i\}_{i=1}^M$ be the set of the unlabeled samples and $\mathcal{P} = \{[b_0, b_1], [b_1, b_2], \ldots\}$ be the partition returned by the pre-processing step given by Algorithm 6. Let $\mathcal{D}_J$ be the distribution of a random variable $(X, Y) \sim \mathcal{D}$ conditioned on the event $\{X \in I_J\}$. Similarly, let $\mathcal{D}_{\mathsf{sh}}$ and $\mathcal{D}_{\mathsf{lg}}$ be the conditional distribution of $\mathcal{D}$ on intervals belonging to $\mathcal{P}_{\mathsf{lg}}$ and $\mathcal{P}_{\mathsf{sh}}$ respectively. Let p_J denote the probability of a point sampled from distribution $\mathcal{D}$ lying in interval I_J. Going forward, we use the shorthand 'probability of interval I_J' to denote p_J. Let p_{lg} and p_{sh} be the probability of set of long and short intervals respectively. Recall that $f^*_{\mathcal{D}}$ is the function which minimizes $\mathrm{err}_{\mathcal{D}}(f)$ for $f \in \mathcal{F}_L$ and $f^*_{\mathcal{D}_J}$ is the function which minimizes this error with respect to the conditional distribution $\mathcal{D}_J$. Let M_J denote the number of unlabeled samples of S lying in interval I_J. For any interval I_J, let $\hat{f}_{S \cap I_J}$ be the ERM with respect to that interval.

Lipschitzness of $\tilde{f}$. For x belonging to intervals say I_J where $I_J^j = [b^j_{J^j-1}, b^j_{J^j}], j \in [d]$ where $I_J \in \mathcal{P}_{\mathsf{sh}}$, let $f_J(x)$ be defined by the output of the Algorithm 6. Now, let $\tilde{f} : [0,1]^d \to [0,1] = f_J(x)$ if $x \in I_J$ be disjoint union of the functions f_J defined on disjoint intervals I_J. Now, we will argue that $\tilde{f}(x)$ is L Lipschitz for all $x \in [0,1]^d$. The function $\tilde{f}(x)$ for each long interval is L Lipschitz by construction. For each long interval I_J where $I_J^j = [b^j_{J^j-1}, b^j_{J^j}] \; \forall j \in [d]$, consider the extension interval $\hat{I}_J$ as defined in Definition 13. The function $\tilde{f}(x)$ is constrained to be 1 for the middle point of the short interval along each dimension i.e. $\tilde{f}(x) = 1 \; \forall x \in [0,1]^d$ if $\exists i \in$

$[d]$ such that $x_i = \frac{b^i_{j-1}+b^i_j}{2}$ where $[b^i_{j-1}, b^i_j]$ is a short interval for dimension i. Note that these middle points of short intervals are precisely the only points of intersection between extensions of two different long intervals because long intervals in each dimension are separated by a distance of $\frac{2}{L}$ by construction and the extension intervals extend up to a shell of thickness around $\frac{1}{L}$ in each dimension. Now, for a query point x belonging to extension interval $\hat{I}_J$ for a long interval I_J takes the value $f(x)$ where $f(x)$ is the Lipschitz extension of the Lipschitz function f_J to the superset $\hat{I}_J$ and constrained to be 1 at the boundary of the shell. Hence, if we can prove that such a Lipschitz extension exists, we can see that the combined function is L Lipschitz. By McShane (1934) (also stated as Theorem 11 in this paper), we can see that such a Lipschitz extension always exists because for point $x \in I_J$ and y belonging to the boundary of the shell, we get that $||x - y||_\infty \geq \frac{1}{L}$ and $|f(x) - f(y)| \leq 1$ and hence, the function f is L-Lipschitz. For query points x which do not belong to any of these extension intervals, we output 1. Basically, these points belong to the short intervals at the boundary of the domain and equivalent to learning Lipschitz extension with empty long interval and constrained to be 1 at the boundary. Hence, the function takes 1 everywhere in this interval. Hence, we can see that the function $\tilde{f}$ is L-Lipschitz.

Error Guarantees for $\tilde{f}$. Now looking at the error rate of the function $\tilde{f}(x)$ and following a repeated application of tower property of expectation, we get that

$$\begin{aligned}\Delta_{\mathcal{D}}(\tilde{f}, \mathcal{F}_L) &= p_{\mathsf{lg}} \Delta_{\mathcal{D}_{\mathsf{lg}}}(\tilde{f}, f^*_{\mathcal{D}}) + p_{\mathsf{sh}} \Delta_{\mathcal{D}_{\mathsf{sh}}}(\tilde{f}, f^*_{\mathcal{D}}) \\ &= \sum_{J : I_J \in \mathcal{P}_{\mathsf{lg}}} p_J \Delta_{\mathcal{D}_J}(\tilde{f}, f^*_{\mathcal{D}}) + p_{\mathsf{sh}} \Delta_{\mathcal{D}_{\mathsf{sh}}}(\tilde{f}, f^*_{\mathcal{D}}) \end{aligned} \tag{10}$$

Now, we need to argue about the error bounds for both long and short intervals to argue about the total error of the function $\tilde{f}$.

Error for short intervals. The probability of short intervals p_{sh} is small with high probability since the total length of short intervals is 2ϵ and the intervals are chosen uniformly randomly. More formally, from Lemma 17, we know that with probability at least $1 - \delta$, the probability of short intervals p_{sh} is upper bounded by $2\epsilon/\delta$. Also, the error for any function f is bounded between $[0, 1]$ since the function's range is $[0, 1]$. Hence, we get that

$$p_{\mathsf{sh}} \Delta_{\mathcal{D}_{\mathcal{P}_{\mathsf{sh}}}}(\tilde{f}, f^*_{\mathcal{D}}) \leq \frac{2\epsilon}{\delta} \tag{11}$$

Error for long intervals: We further divide the long intervals into 3 subtypes:

$$\begin{aligned}\mathcal{P}_{\mathsf{lg},1} &:= \left\{ I_J \mid I_J \in \mathcal{P}_{\mathsf{lg}},\ p_J \geq \frac{\epsilon}{(\frac{L\epsilon}{d})^d},\ M_J \geq \frac{1}{\epsilon^2}(\Omega(\frac{d}{\epsilon^2}))^d) \log(\frac{1}{\epsilon}) \right\}, \\ \mathcal{P}_{\mathsf{lg},2} &:= \left\{ I_J \mid I_J \in \mathcal{P}_{\mathsf{lg}},\ p_J \geq \frac{\epsilon}{(\frac{L\epsilon}{d})^d},\ M_J < \frac{1}{\epsilon^2}(\Omega(\frac{d}{\epsilon^2}))^d) \log(\frac{1}{\epsilon}) \right\}, \\ \mathcal{P}_{\mathsf{lg},3} &:= \left\{ I_J \mid I_J \in \mathcal{P}_{\mathsf{lg}},\ p_J < \frac{\epsilon}{(\frac{L\epsilon}{d})^d} \right\}. \end{aligned}$$

The intervals in both first and second types have large probability p_J with respect to distribution $\mathcal{D}$ but differ in the number of unlabeled samples in S lying in them. Finally, the intervals in third subtype $\mathcal{P}_{\mathsf{lg},3}$ have small probability p_J with respect to distribution $\mathcal{D}$. Now, we can divide the total error of long intervals into error in these subtypes

$$\sum_{J:I_J\in\mathcal{P}_{\mathsf{lg}}} p_J\Delta_{\mathcal{D}_J}(\tilde{f}, f^*_{\mathcal{D}}) = \underbrace{\sum_{J:I_J\in\mathcal{P}_{\mathsf{lg},1}} p_J\Delta_{\mathcal{D}_J}(\tilde{f}, f^*_{\mathcal{D}})}_{E1} + \underbrace{\sum_{J:I_J\in\mathcal{P}_{\mathsf{lg},2}} p_J\Delta_{\mathcal{D}_J}(\tilde{f}, f^*_{\mathcal{D}})}_{E2} + \underbrace{\sum_{J:I_J\in\mathcal{P}_{\mathsf{lg},3}} p_J\Delta_{\mathcal{D}_J}(\tilde{f}, f^*_{\mathcal{D}})}_{E3} \tag{12}$$

Now, we will argue about the contribution of each of the three terms above.

<u>Bounding $E3$.</u> Since there are at most $(\frac{L\epsilon}{d})^d$ long intervals and each of these intervals I_J has probability p_J upper bounded by $\frac{\epsilon}{(\frac{L\epsilon}{d})^d}$, the total probability combined in these intervals is at most ϵ. Also, in the worst case, the loss can be 1. Hence, we get an upper bound of ϵ on E_3.

<u>Bounding $E2$.</u> From Lemma 16, we know that with success probability δ, these intervals have total probability upper bounded by ϵ/δ. Again, the loss can be 1 in the worst case. Hence, we can get an upper bound of ϵ/δ on E_2.

<u>Bounding $E1$.</u> Let F_J denote the event that $\Delta_{\mathcal{D}_J}(\hat{f}_{S\cap I_J}, f^*_{\mathcal{D}_J}) > \epsilon$. The expected error of intervals I_J in $\mathcal{P}_{\mathsf{lg},1}$ is then

$$\begin{aligned}\mathbb{E}[\sum_{J:I_J\in\mathcal{P}_{\mathsf{lg},1}} p_J\Delta_{\mathcal{D}}(\tilde{f}, f^*_{\mathcal{D}_J})] &\overset{\text{(i)}}{\leq} \mathbb{E}[\sum_{J:I_J\in\mathcal{P}_{\mathsf{lg},1}} p_J\Delta_{\mathcal{D}_J}(\hat{f}_{S\cap I_J}, f^*_{\mathcal{D}_J})] \\ &= \sum_{J:I_J\in\mathcal{P}_{\mathsf{lg},1}} p_J(\mathbb{E}[\Delta_{\mathcal{D}_J}(\hat{f}_{S\cap I_J}, f^*_{\mathcal{D}_J})|F_J]\Pr(F_J) + \mathbb{E}[\Delta_{\mathcal{D}_J}(\hat{f}_{S\cap I_J}, f^*_{\mathcal{D}_J})|\neg F_J]\Pr(\neg F_J)) \\ &\overset{\text{(ii)}}{\leq} \sum_{J:I_J\in\mathcal{P}_{\mathsf{lg},1}} p_J(1\cdot\epsilon + \epsilon\cdot 1) \leq 2\epsilon,\end{aligned}$$

where step (i) follows by noting that $\tilde{f} = \hat{f}_{S\cap I_J}$ for all long intervals $I_J \in \mathcal{P}_{\mathsf{lg}}$ and that $f^*_{\mathcal{D}_J}(x)$ is the minimizer of the error $\mathrm{err}_{\mathcal{D}_J}(f)$ over all L-Lipschitz functions, and step (ii) follows since $\mathbb{E}[\Delta_{\mathcal{D}_J}(\hat{f}_{S\cap I_J}, f^*_{\mathcal{D}_J})|\neg F_J] \leq \epsilon$ by the definition of event F_J and $\Pr(F_J) \leq \epsilon$ follows from a standard uniform convergence argument (detailed in Lemma 18). Now, using Markov's inequality, we get that $E_1 \leq 2\epsilon/\delta$ with failure probability at most δ.

Plugging the error bounds obtained in equations (11) and (12) into equation (10) and setting $\delta = \frac{1}{20}$ establishes the required claim.

Label Query Complexity. Observe that for any given query point x^*, $\tilde{f}(x^*)$ can be computed by only querying the labels of the interval in which x lies (in case if x^* lies in a long interval) or extension of the interval in which x^* lies (in case if x^* lies in a short interval) and hence, would only require $\frac{1}{\epsilon^2}(O(\frac{d}{\epsilon^2}))^d\log(\frac{1}{\epsilon})$ active label queries over the set S of $\frac{1}{\epsilon^3}(O(\frac{L}{\epsilon}))^d\log(\frac{1}{\epsilon})$ unlabeled samples. ■

Now, we will state the sample complexity bounds for estimating the error of the optimal function amongst the class of Lipschitz functions up to an additive error of ϵ.

Theorem 15 *For any distribution $\mathcal{D}$ over $[0,1]^d \times [0,1]$, Lipschitz constant $L > 0$ and parameter $\epsilon \in [0,1]$, Algorithm 3 uses $\frac{1}{\epsilon^4}(O(\frac{d}{\epsilon^2}))^d\log(\frac{1}{\epsilon})$ active label queries on $\frac{1}{\epsilon^3}(O(\frac{L}{\epsilon}))^d\log(\frac{1}{\epsilon})$ unlabeled samples from distribution $\mathcal{D}_x$ and produces an output $\widehat{err}_{\mathcal{D}}(\mathcal{F}_L)$ satisfying*

$$|\widehat{err}_{\mathcal{D}}(\mathcal{F}_L) - err_{\mathcal{D}}(\mathcal{F}_L)| \leq \epsilon$$

with probability at least $\frac{1}{2}$.

The proof goes exactly like the proof for the corresponding theorem in the one-dimensional case (Theorem 2).

The following lemma proves that with enough unlabeled samples, a large fraction of long intervals have enough unlabeled samples in them which is eventually used to argue that they will be sufficient to learn a function which is approximately close to the optimal Lipschitz function over that interval.

Lemma 16 *For any distribution $\mathcal{D}_x$, consider a set $S = \{x_1, x_2, \cdots, x_M\}$ of unlabeled samples where each sample $x_i \overset{i.i.d.}{\sim} \mathcal{D}_x$. Let $\mathcal{G}$ be the set of long intervals $\{I_i\}$ each of which satisfies $p_J = \Pr_{x \sim \mathcal{D}_x}(x \in I_J) \geq \frac{\epsilon}{(\frac{L\epsilon}{d})^d}$. Let E_J denote the event that $\sum_{x_j \in S} \mathbb{I}[x_j \in I_J] < \frac{1}{\epsilon^2}(\frac{d}{\epsilon^2})^d \log(\frac{1}{\epsilon})$. Then, we have*

$$\sum_{I_J \in \mathcal{G}} p_J \mathbb{I}[E_J] \leq \frac{\epsilon}{\delta}$$

with failure probability atmost δ for $M = \frac{1}{\epsilon^3}(\Omega(\frac{L}{\epsilon}))^d \log(\frac{1}{\epsilon})$.

Proof For any interval $I \in \mathcal{G}$, we have that $\mathbb{E}[\sum_{x_j \in S} \mathbb{I}[x_j \in I]] \geq \frac{c_d}{\epsilon^2}(\frac{d}{\epsilon^2})^d \log(\frac{1}{\epsilon})$ for some constant c_d depending on the dimension d. Using Hoeffding inequality, we can get that $\Pr(E_i) \leq \epsilon$ for all intervals $I_i \in \mathcal{G}$. Calculating expectation of the desired quantity, we get

$$\mathbb{E}[\sum_{I_J \in \mathcal{G}} p_J \mathbb{I}[E_J]] = \sum_{I_J \in \mathcal{G}} p_J \Pr[E_J] \leq \epsilon \sum_{I_J \in \mathcal{G}} p_J \leq \epsilon$$

We get the desired result using Markov's inequality. ■

The following lemma states that the probability of short intervals p_{sh} is small with high probability. Consider the case of uniform distribution. In this case, since the short intervals cover only 2ϵ fraction of the domain $[0,1]^d$, their probability p_{sh} is upper bounded by 2ϵ. The case for arbitrary distributions holds because the intervals are chosen randomly.

Lemma 17 *When we divide the $[0,1]^d$ domain into long and short intervals as in the preprocessing step (Algorithm 5), then*

$$p_{\mathsf{sh}} = \sum_{I_J \in \mathcal{P}_{\mathsf{sh}}} p_J \leq \frac{2\epsilon}{\delta}$$

with failure probability atmost δ.

Proof Now, we consider the division of every dimension i.e. $[0,1]$ independently into alternating intervals of length $\frac{d}{L\epsilon}$ and $\frac{2}{L}$ with the offset chosen uniformly randomly from $\{0, 1, 2, \cdots, \frac{d}{2\epsilon}\}\frac{2}{L}$. For any set of fixed offsets chosen for the $d-1$ dimensions, the intervals of length $\frac{2}{L}$ chosen for the dth dimension combined are disjoint in each of these divisions and together cover the entire $[0,1]^d$ and hence amount to probability mass 1. Therefore, the total probability mass covered in the total $(\frac{d}{2\epsilon}+1)^d$ possible divisions is $d(\frac{d}{2\epsilon}+1)^{d-1}$. Hence, there are at most δ fraction out of the total $(\frac{d}{2\epsilon}+1)^d$ cases where the short intervals have probability greater than $\frac{2\epsilon}{\delta}$. Hence, with probability $1-\delta$, the short intervals have probability upper bounded by $\frac{2\epsilon}{\delta}$. ■

Lemma 18 *Let $I_i \in \mathcal{P}_{\mathsf{lg},1}$ be any long interval of subtype 1. For the event $F_i = \{\Delta_{\mathcal{D}_i}(\hat{f}_{S\cap I_i}, f^*_{\mathcal{D}_i}) > \epsilon\}$, we have*

$$\Pr(F_i) \leq \epsilon$$

Proof We know that the covering number for the class of d-dimensional L-Lipschitz functions supported on the interval $[0, l]$ is $(O(Ll/\epsilon))^d$ (Lemma 12). For, Lipschitz functions supported on long intervals of length $l = d/(L\epsilon)$ along each dimension, we get this complexity as $(O(d/\epsilon^2))^d$. We know by standard results in uniform convergence, that the number of samples required for uniform convergence up to an error of ϵ and failure probability δ for all functions in a class $\mathcal{F}_L$ is $\Omega(((\text{Covering number of } \mathcal{F}_L)/\epsilon^2)\log(1/\delta))$ (Lemma 10) and hence, for long intervals we get this estimate as $\frac{1}{\epsilon^2}(\Omega(\frac{d}{\epsilon^2}))^d \log(\frac{1}{\delta})$. Given that there are $\frac{1}{\epsilon^2}(\Omega(\frac{d}{\epsilon^2}))^d \log(\frac{1}{\epsilon})$ samples in these intervals by design, the function learned using these samples is only additively worse than the function with minimum error using standard learning theory arguments with failure probability at most ϵ. Thus, we get that

$$\Delta_{\mathcal{D}_i}(f_{S\cap I_i}, f^*_{\mathcal{D}_i}) \leq \epsilon \quad \forall I_i \in \mathcal{P}_{\mathsf{lg},1}$$

with failure probability at most ϵ. ■

Appendix C. Nadaraya-Watson Estimator

In this section, we state the proof of Theorem 4 and the lemmas needed. But, first we state a theorem from Backurs et al. (2018) which will be crucial in the proof. The following theorem essentially states that for certain nice kernel $K(x, y)$, it is possible to estimate $\frac{1}{N}\sum_{i=1}^{N} K(x, x_i)$ with a multiplicative error of ϵ for any query x efficiently. In this section, we will use $A \preceq B$ (or $A \succeq B$) to denote that $B - A$ (or $A - B$) is a positive semidefinite matrix for two positive semidefinite matrices A and B.

Theorem 19 (Theorem 11 of Backurs et al. (2018)) *There exists a data structure that given a data set $P \subset \mathbb{R}^d$ of size N, using $O(dL2^{O(t)}\log(\frac{\Phi N}{\delta}))\frac{1}{\epsilon^2} \cdot N$ space and preprocessing time, for any (L, t) nice kernel and a query $q \in \mathbb{R}^d$, estimates $KDF_P(q) = \frac{1}{|P|}\sum_{y\in P} k(q, y)$ with accuracy $(1 \pm \epsilon)$ in time $O(dL2^{O(t)}\log(\frac{\Phi N}{\delta}))\frac{1}{\epsilon^2})$ with probability at least $1 - \frac{1}{poly(N)} - \delta$.*

The kernel used in this setting $K_A(x, y) = \frac{1}{1+||A(x-y)||_2^2}$ $\forall A \in \mathcal{A}$ is $(4, 2)$ smooth according to their definition of smoothness as shown in Definition 1 in Backurs et al. (2018) and hence can be computed efficiently. Moreover, as mentioned in Backurs et al. (2018), it is also possible to remove the dependence on the aspect ratio and in turn achieve time complexity of $O(\frac{d}{\epsilon^2}\log(\frac{n}{\delta}))$ with a preprocessing time of $O(\frac{dN}{\epsilon^2}\log(\frac{N}{\delta}))$. Note that the data structure in Backurs et al. (2018) only depends on the smoothness properties of the kernel and hence, the same data structure can be used for simultaneously computing the kernel density for all kernels K_A for all $A \in \mathcal{A}$. As a direct corollary of Theorem 19, we obtain that it is possible to efficiently estimate $p_{S,A}(q, x_i)$ $\forall x_i \in S, q \in \mathbb{R}^d, A \in \mathcal{A}$ since multiplication by a constant still preserves the multiplicative approximation (Corollary 3). Now, we restate Theorem 4 and its proof.

Theorem 4 *For a d-dimensional unlabeled pointset S with $|S| = N$, Algorithm 4 queries $O(\frac{1}{\epsilon^2}(d\log(\frac{1}{\epsilon}) + \log(\frac{1}{\delta}))$ labels from S and outputs $\hat{L}_S$ such that*

$$|\hat{L}_S - \min_{A\in\mathcal{A}} \mathop{\mathbb{E}}_{x\sim\mathcal{D}} \sum_i p_{S,A}(x_i, x)|f(x_i) - f(x)|| \leq \epsilon$$

with a failure probability of at most $\delta + \frac{d}{\epsilon^{d+2} poly(N)} \log(\frac{1}{\epsilon\delta})$ *and runs in time* $\tilde{O}(\frac{d^2}{\epsilon^{d+4}} + \frac{dN}{\epsilon^2})$.

Proof Let us assume that A^* is the optimal matrix which minimizes the prediction error i.e.

$$A^* = \operatorname*{argmin}_{A \in \mathcal{A}} \mathop{\mathbb{E}}_{x \sim \mathcal{D}} \sum_i p_{S,A}(x_i, x) |f(x_i) - f(x)|$$

Let us consider a set $\mathcal{A}_\epsilon$, an ϵ-covering of the set of matrices $\mathcal{A}$ with size $T = |\mathcal{A}_\epsilon| = O(\frac{1}{\epsilon^d})$, that is,

$$\mathcal{A}_\epsilon = \{A \in \mathbb{R}^{d \times d} \mid A_{i,j} = 0 \; \forall \; i \neq j \text{ and } A_{i,i} \in \{1, 1+\epsilon, (1+\epsilon)^2, \cdots, 2\} \; \forall i \in [d]\}.$$

From Lemma 20, we know that it is sufficient to estimate $\min_{A \in \mathcal{A}_\epsilon} \mathbb{E}_{x \sim \mathcal{D}} | \sum_i p_{S,A}(x_i, x) |f(x_i) - f(x)|$ up to an error of ϵ because the optimal error in $\mathcal{A}$ and $\mathcal{A}_\epsilon$ are within ϵ of each other. To estimate this, we use the estimator from Algorithm 4.

Now, we will prove that the estimator is within $E_{x \sim \mathcal{D}} \sum_{i=1}^N p_{S,A}(x, x_i) |f(x_i) - f(x)| \pm \epsilon$ with high probability for all $A \in \mathcal{A}_\epsilon$. Let E be the event that the estimators $\hat{p}_{S,A}$ from Algorithm 4 all approximate $p_{S,A}$ up to a multiplicative error of ϵ, that is,

$$E = \{\hat{p}_{S,A}(z_i, \tilde{z}_i) \in p_{S,A}(z_i, \tilde{z}_i)[1-\epsilon, 1+\epsilon] \quad \forall i \in [M], \forall A \in \mathcal{A}_\epsilon\}.$$

Now, we will break the probability of the estimator $\hat{L}_{S,K_A}$ not being within ϵ close to the true value L_{S,K_A} for all $A \in \mathcal{A}$.

$$\Pr(|\hat{L}_{S,K_A} - L_{S,K_A}| > \epsilon) = \underbrace{\Pr(|\hat{L}_{S,K_A} - L_{S,K_A}| > \epsilon | E)}_{T_1} \Pr(E) + \Pr(|\hat{L}_{S,K_A} - L_{S,K_A}| > \epsilon | \neg E) \underbrace{\Pr(\neg E)}_{T_2}$$

Bounding T_1. Computing the expectation of the estimator conditioned on the event E, we get that

$$\begin{aligned}
E[\hat{L}_{S,K_A} | E] &= \frac{1}{M} \sum_i E_{z_i \sim \mathcal{D}, \tilde{z}_i \sim p_{S,I}(z_i, \tilde{z}_i)} [\frac{\hat{p}_{S,A}(z_i, \tilde{z}_i)}{p_{S,I}(z_i, \tilde{z}_i)} |f(z_i) - f(\tilde{z}_i)|] \\
&= \frac{1}{M} \sum_i E_{z_i \sim \mathcal{D}} [\sum_{j=1}^N \hat{p}_{S,A}(z_i, x_j) |f(z_i) - f(x_j)|] \\
&= E_{z \sim \mathcal{D}} \sum_{j=1}^N \hat{p}_{S,A}(z_i, x_j) |f(z_i) - f(x_j)| \\
&\in E_{z \sim \mathcal{D}} \sum_{j=1}^N p_{S,A}(z_i, x_j) |f(z_i) - f(x_j)| [1-\epsilon, 1+\epsilon] \\
&\in L_{S,K_A}[1-\epsilon, 1+\epsilon]
\end{aligned}$$

Now, we will show that the estimator is close to its expectation with high probability conditioned on the event E. Since we know that $|f(z_i) - f(\tilde{z}_i)| \leq 1$ and

$$p_{S,A}(z_i, \tilde{z}_i) \leq 16 p_{S,I}(z_i, \tilde{z}_i) \quad \forall z_i, \tilde{z}_i \in \mathbb{R}^d, \forall A \in \mathcal{A},$$

from Lemma 21, we get that each entry of our estimator is bounded between $[0, 16(1+\epsilon)]$. Hence, using Hoeffding's inequality we get that

$$\Pr(|\hat{L}_{S,K_A} - E[\hat{L}_{S,K_A}|E]| \geq \epsilon | E) \leq 2e^{\frac{-2M\epsilon^2}{256(1+\epsilon)^2}}.$$

Hence, for $M = O(\frac{1}{\epsilon^2}\log(\frac{T}{\delta}))$ and union bound over the ϵ cover of size T, we get $T_1 \leq \delta$

Bounding T_2. Using Corollary 3 with δ as $\frac{\delta\epsilon^d}{M}$ and a union bound over the $M \cdot \frac{1}{\epsilon^d}$ computations of $\hat{p}_{S,A}$, we get that

$$\Pr(\neg E) \leq \delta + \frac{1}{poly(N)}\frac{M}{\epsilon^d}$$

Combining the upper bounds for T_1 and T_2,

$$\Pr(|\hat{L}_{S,K_A} - L_{S,K_A}| > \epsilon) \leq 2\delta + \frac{1}{poly(N)}\frac{M}{\epsilon^d},$$

for $M = O(\frac{1}{\epsilon^2}\log(\frac{T}{\delta}))$. Substituting $T = \frac{1}{\epsilon^d}$, we get a sample complexity of $M = O(\frac{1}{\epsilon^2}(d\log(\frac{1}{\epsilon}) + \log(\frac{1}{\delta})))$ and failure probability of $2\delta + \frac{1}{poly(N)\epsilon^{d+2}}(d\log(\frac{1}{\epsilon}) + \log(\frac{1}{\delta}))$.

Running Time. For the running time, since we have a total of $O(\frac{1}{\epsilon^2}(d\log(\frac{1}{\epsilon}) + \log(\frac{1}{\delta}))$ samples and we have to take a sum over these samples for each $A \in \mathcal{A}_\epsilon$, we get a total running time of $O(\frac{1}{\epsilon^d}(\frac{1}{\epsilon^2}(d\log(\frac{1}{\epsilon}) + \log(\frac{1}{\delta})))$ for this part. $p_{S,I}$ needs to be computed only once for each sample and this leads to a running time of $O(N(\frac{1}{\epsilon^2}(d\log(\frac{1}{\epsilon}) + \log(\frac{1}{\delta})))$. We also compute an estimator $\hat{p}_{S,A}$ for each of the sample pair for each $A \in \mathcal{A}_\epsilon$ and thus from Corollary 3, we get a preprocessing time of $O(\frac{Nd}{\epsilon^2}\log(\frac{NM}{\delta\epsilon^d}))$ and $O(\frac{Md}{\epsilon^{d+2}}\log(\frac{NM}{\delta\epsilon^d}))$ time in computing the estimator for each $A \in \mathcal{A}_\epsilon$. This completes the proof of the theorem. ■

The following lemma used in the proof of Theorem 4 states that it is sufficient to consider an ϵ-net of the set of matrices $\mathcal{A}$ to approximate the minimum error up to an additive error of ϵ.

Lemma 20 *Let us consider a set $\mathcal{A}_\epsilon$, an ϵ-covering of the set of matrices $\mathcal{A}$ i.e.*

$$\mathcal{A}_\epsilon = \{A \in \mathbb{R}^{d\times d} \mid A_{i,j} = 0 \,\forall\, i \neq j \text{ and } A_{i,i} \in \{1, 1+\epsilon, (1+\epsilon)^2, \cdots, 2\} \,\forall i \in [d]\}$$

Then the minimum prediction error for $A \in \mathcal{A}_\epsilon$ additively approximates the minimum prediction error for $A \in \mathcal{A}$ i.e.

$$\Big|\min_{A\in\mathcal{A}_\epsilon} \mathop{\mathbb{E}}_{x\sim\mathcal{D}} \sum_i p_{S,A}(x_i, x)|f(x_i) - f(x)| - \min_{A\in\mathcal{A}} \mathop{\mathbb{E}}_{x\sim\mathcal{D}} \sum_i p_{S,A}(x_i, x)|f(x_i) - f(x)|\Big| \leq 15\epsilon$$

Proof We first show that that if $(1+\epsilon)^{-1}A_2 \preceq A_1 \preceq A_2(1+\epsilon)$, then

$$\Big|\mathop{\mathbb{E}}_{x\sim\mathcal{D}} \sum_i p_{S,A_1}(x_i, x)|f(x_i) - f(x)| - \mathop{\mathbb{E}}_{x\sim\mathcal{D}} |\sum_i p_{S,A_2}(x_i, x)|f(x_i) - f(x)|\Big| \leq 15\epsilon$$

Note that the loss difference can also be written as $\mathbb{E}_{x\sim\mathcal{D}} \sum_i |p_{S,A_1}(x_i, x) - p_{S,A_2}(x_i, x)||f(x_i) - f(x)|$. Now from Lemma 21, we know that if $(1+\epsilon)^{-1}A_2 \preceq A_1 \preceq A_2(1+\epsilon)$, then $(1+$

$\epsilon)^{-4}p_{S,A_2}(x_i,x) \leq p_{S,A_1}(x_i,x) \leq p_{S,A_2}(x_i,x)(1+\epsilon)^4$. Using this, we get that

$$\begin{aligned}
&E_{x\sim\mathcal{D}}\sum_i |p_{S,A_1}(x_i,x) - p_{S,A_2}(x_i,x)||f(x_i) - f(x)| \\
&\qquad\qquad\qquad\qquad \leq E_{x\sim\mathcal{D}}\sum_i |p_{S,A_2}(x_i,x)(1+\epsilon)^4 - p_{S,A_2}(x_i,x)||f(x_i) - f(x)| \\
&\qquad\qquad\qquad\qquad \leq 15\epsilon E_{x\sim\mathcal{D}}\sum_i |p_{S,A_2}(x_i,x)||f(x_i) - f(x)| \\
&\qquad\qquad\qquad\qquad \leq 15\epsilon E_{x\sim\mathcal{D}}\sum_i |p_{S,A_2}(x_i,x)| \leq 15\epsilon
\end{aligned}$$

Hence, the loss difference is also bounded by 15ϵ. Now, we will prove that $\exists A \in \mathcal{A}_\epsilon$ such that $(1+\epsilon)^{-1}A \preceq A^* \preceq A(1+\epsilon)$ where $A^* = \operatorname{argmin} \mathbb{E}_{x\sim\mathcal{D}}\sum_i |p_{S,A}(x_i,x)||f(x_i) - f(x)|$. This will be sufficient to show that the minimum error for the set $\mathcal{A}_\epsilon$ and $\mathcal{A}$ differ from each other by an additive error at most ϵ. Consider each of the diagonal entries of A to be the value in $\{1, 1+\epsilon, (1+\epsilon)^2, \cdots, 2\}$ closest to the corresponding entry of A^*. It is easy to see that $(1+\epsilon)^{-1}A \preceq A^* \preceq A(1+\epsilon)$. ■

The following lemma states that if two matrices A_1 and A_2 are multiplicatively close to each other in terms of all their eigenvalues, then the corresponding probabilities for any query point x and any data point x_i are also multiplicatively close to each other.

Lemma 21 *For any two matrices $A_1, A_2 \in \mathbb{R}^{d\times d}$, if $\frac{1}{1+\epsilon}A_2 \preceq A_1 \preceq A_2(1+\epsilon)$, then*

$$\frac{1}{(1+\epsilon)^4}p_{S,A_2}(x_i,x) \leq p_{S,A_1}(x_i,x) \leq p_{S,A_2}(x_i,x)(1+\epsilon)^4 \quad \forall x \in \mathbb{R}^d, x_i \in S.$$

Proof Using $(1+\epsilon)^{-1}A_2 \preceq A_1 \preceq A_2(1+\epsilon)$, we get that

$$\begin{aligned}
||A_2(x_i - x)||\frac{1}{1+\epsilon} \quad &\leq \quad ||A_1(x_i - x)|| \quad \leq \quad ||A_2(x_i - x)||(1+\epsilon) \\
1 + ||A_2(x_i - x)||^2\frac{1}{(1+\epsilon)^2} \quad &\leq 1 + ||A_1(x_i - x)||^2 \leq \quad 1 + ||A_2(x_i - x)||^2(1+\epsilon)^2 \\
\frac{1}{(1+\epsilon)^2}(1 + ||A_2(x_i - x)||^2) &\leq 1 + ||A_1(x_i - x)||^2 \leq (1 + ||A_2(x_i - x)||^2)(1+\epsilon)^2 \\
\frac{1}{(1+\epsilon)^2}K_{A_2}(x_i,x) \quad &\leq \quad K_{A_1}(x_i,x) \quad \leq \quad K_{A_2}(x_i,x)(1+\epsilon)^2 \qquad (13)
\end{aligned}$$

Hence, using the inequality in equation 13, we get that

$$\begin{aligned}
\frac{1}{(1+\epsilon)^4}\frac{K_{A_2}(x_i,x)}{\sum_{x_i\in S}K_{A_2}(x_i,x)} &\leq \frac{K_{A_1}(x_i,x)}{\sum_{x_i\in S}K_{A_1}(x_i,x)} \leq \frac{K_{A_2}(x_i,x)}{\sum_{x_i\in S}K_{A_2}(x_i,x)}(1+\epsilon)^4 \\
\frac{1}{(1+\epsilon)^4}p_{S,A_2}(x_i,x) \quad &\leq \quad p_{S,A_1}(x_i,x) \quad \leq \quad p_{S,A_2}(x_i,x)(1+\epsilon)^4
\end{aligned}$$

This completes the proof of the lemma. ■