JMLR: Workshop and Conference Proceedings vol 30 (2013) 1–22

PLAL: Cluster-based Active Learning

Ruth Urner RURNER@CS.UWATERLOO.CA
School of Computer Science, University of Waterloo, Canada, ON, N2L 3G1

Sharon Wulff SHARON.WULFF@INF.ETHZ.CH
Department of Computer Science, ETH, Zurich, Switzerland

Shai Ben-David SHAI@CS.UWATERLOO.CA
School of Computer Science, University of Waterloo, Canada, ON, N2L 3G1

Abstract

We investigate the label complexity of active learning under some smoothness assumptions on the data-generating process. We propose a procedure, PLAL, for "activising" passive, sample-based learners. The procedure takes an unlabeled sample, queries the labels of some of its members, and outputs a full labeling of that sample. Assuming the data satisfies "Probabilistic Lipschitzness", a notion of clusterability, we show that for several common learning paradigms, applying our procedure as a preprocessing leads to provable label complexity reductions (over any "passive" learning algorithm, under the same data assumptions). Our labeling procedure is simple and easy to implement. We complement our theoretical findings with experimental validations.

Keywords: learning theory, agnostic active learning, label complexity

1. Introduction

Traditional machine learning theory focuses on the fully supervised setup, in which a learner has access to a labeled randomly generated training sample. However, in many learning applications labeling training examples is expensive. In Active Learning (AL), the learner gets a sample of unlabeled instances, and can choose which labels to obtain based on the instances and perhaps previously seen labels. By choosing the instances wisely, the learner aims to achieve good prediction performance while using as little labeled instances as possible, and, in particular, less than the amount required for learning from a fully labeled random sample. Active learning paradigms are successful in practice and there is also a variety of theoretical studies analyzing the possibilities and limitations of AL. However, several studies have shown that, under worst-case scenarios, AL algorithms are bound to require as many labeled sample points as their "passive" fully supervised counterparts (Dasgupta, 2005; Kääriäinen, 2006; Beygelzimer et al., 2009). Those negative results set the frame for most of the theoretical research on AL. Rather than trying to show that active choice of label queries can *always* reduce the number of training labels, one aims to identify properties of the learning task under which an AL paradigm is beneficial.

For many label prediction tasks, there is a significant correlation between the (marginal) distribution over the data points and the labels. Under a suitable data representation, or feature choice, we expect that the closer two instances are, the less likely they are to have different labels. Probabilistic Lipschitzness (PL) is a measure that quantifies this

correlation. It can also be viewed as a way to model the *cluster assumption*, which is often invoked in the context of semi supervised learning.

Our algorithm (PLAL) follows a paradigm proposed by Dasgupta and Hsu (2008) for exploiting cluster structure for active learning. While most previous work on the theory of active learning focused on an efficient version space reduction for learning a hypothesis class, Dasgupta and Hsu (2008) suggest a labeling procedure based on a hierarchical clustering of the training data. The authors show that assuming that the learner is given a "good" hierarchical clustering, an unlabeled sample can be labeled almost correctly with relatively few label queries. They suggest to then feed the now labeled sample to any standard learning procedure. In this paper we analyze a version of their approach under the assumption of PL. This condition is weaker than the availability of a "good" clustering tree in that we only need it for the analysis of our procedure (as opposed to the need for a successful preprocessing step that finds the good clustering tree). We believe that cluster-based active learning is an important research direction that has not received enough attention from the learning theory community so far.

Our main results show upper bounds for the sample complexity of PLAL-based active learning and lower bounds for the sample complexity of standard (passive) learning under similar assumptions. In particular, we show that under polynomial rates of Probabilistic Lipschitzness, PLAL significantly reduces the sample complexity of some VC-classes. We present similar results for general (Nearest Neighbor based) learning. The PLAL activising procedure is computationally efficient and can be applied to any noise-tolerant sample-based learning algorithm (see Section 5.1).

In Section 3 we present our framework for the sample complexity of learning as well as our notion of Probabilistic Lipschitzness. Our sample labeling procedure, PLAL is presented in Section 4. We show that PLAL assigns incorrect labels to at most a fraction of the original unlabeled sample. Furthermore, we show that, under the assumption of bounded PL, PLAL queries labels only for a small fraction of the input sample points. In Section 5, we analyze under which conditions the sample labeled PLAL can be used to reduce the label complexity of a learning algorithm and present our main results. We present some experimental results on our labeling procedure in the appendix.

2. Related Work

The survey "Two faces of active learning" by Dasgupta (2011), contrasts two general approaches for active learning: Using label queries to more efficiently search through a hypothesis space and exploiting cluster structure in data. Almost all of the theoretical work so far has focused on the first setup. Starting with Dasgupta (2004) there is a large body of work that analyzes these paradigms in the realizable case and under separability with a margin (e.g. Balcan et al. (2007), Balcan et al. (2010), Gonen et al. (2011)). There have been extensive efforts to generalize the positive results for active learning from the realizable to the *agnostic* case. Lower bounds of $\Omega(1/\epsilon^2)$ by Kääriäinen (2006) and Beygelzimer et al. (2009) imply that, again, improvements in label complexity for learning a hypothesis class are not possible in general. Thus, here as well, research focuses on identifying parameters that characterize learning tasks where active learning is beneficial. The, so far, most prominent such parameter is *disagreement coefficient*, introduced by Hanneke (2007). It

has been used to bound the label complexity of various querying strategies (Hanneke, 2007; Dasgupta et al., 2008; Beygelzimer et al., 2010, 2009). However, the bounds on the number of label queries in these papers all involve the approximation error of the hypothesis class. They become relevant only when the approximation error is small.

A first approach at exploiting cluster structure by active learning has been presented by Dasgupta and Hsu (2008). That paper proposes a labeling strategy for an unlabeled dataset, when the learner is also given a hierarchical clustering of the data. A bound on the number of label queries provided in this work depends on the depth of the effectively used clustering tree; however, it is unclear how to control this parameter. Our work builds on the ideas of Dasgupta and Hsu (2008). We provide a rigorous analysis of the labeling strategy and show how to use it for the second round of learning in a way that admits provable performance guarantees and reduction of label complexity under the assumption of Probabilistic Lipschitzness. A version of the PL parameter was introduced by Steinwart and Scovel (2007) under the name *geometric noise exponent*. Steinwart and Christmann (2008) show that when such a parameter (here called *margin exponent*) is combined with bounds on the noise rate and marginal distribution near the decision boundaries of data, it can be used to bound the approximation error of Gaussian kernels for that data. (Urner et al., 2011b) introduced the version employed here and applied it to formally establish benefits of unlabeled data for semi-supervised learning (Urner et al., 2011b). We bound the number of queries of our algorithm in terms of the Probabilistic Lipschitzness of the underlying data distribution (independently of the depth of the resulting cluster tree and independently of the approximation error of a class).

A framework, where an unlabeled sample is labeled by a preliminary, active labeling procedure and then fed to a standard learner has been introduced by Hanneke (2012) under the title "activized learning". Assuming the data is realizable by a VC-class, the author presents labeling procedures based on repeated computations of the shatter function of version spaces and shows how this labeling procedure reduces the label complexity of the original standard learner. Our PLAL procedure achieves label complexity reduction results for data with bounded PL, which we believe is a more realistic assumption, and is substantially simpler and easy to implement.

3. Definitions

Standard (passive) Learning We consider *domain sets* $\mathcal{X} = [0,1]^d$ for some dimension d, and the *label set* $\{0,1\}$. We assume that the data for a learning problem is generated by some distribution P over $\mathcal{X} \times \{0,1\}$. We denote the marginal distribution of P over $\mathcal{X}$ by $P_{\mathcal{X}}$ and let $l : \mathcal{X} \to [0,1]$ denote the induced conditional labeling probability function, $l(x) = P(y = 1|x)$. In this work, we focus on the case that the labeling is deterministic, namely, $l(x) \in \{0,1\}$ for all $x \in \mathcal{X}$. A *hypothesis*, is a function $h : \mathcal{X} \to \{0,1\}$, and a *hypothesis class* is a set of hypotheses. For a hypothesis $h : \mathcal{X} \to \{0,1\}$, we define the *error* of h with respect to P as $\mathrm{Err}_P(h) = \Pr_{(x,y)\sim P}(y \neq h(x))$. For a class H of hypotheses on $\mathcal{X}$, we let the smallest error of a hypothesis $h \in H$ with respect to P (the *approximation error of H with respect to P*) be denoted by $\mathrm{Err}_P(H) := \inf_{h\in H} \mathrm{Err}_P(h)$.

A *learner* $\mathcal{A}$ takes a sample $S = ((x_1, y_1), \ldots, (x_m, y_m))$ and outputs a hypothesis $h : \mathcal{X} \to \{0,1\}$. We define the *empirical error* of a hypothesis as $\mathrm{Err}_S(h) = |\{(x,y) \in S \ : \ y \neq$

$h(x)\}|/|S|$. We say that an algorithm $\mathcal{A}$ *learns* some hypothesis class H over $\mathcal{X}$ with respect to a set of distribution $\mathcal{Q}$ over $\mathcal{X} \times \{0,1\}$, if there exists a function $m : (0,1) \times (0,1) \to \mathbb{N}$ such that, for all distributions $P \in \mathcal{Q}$, with probability at least $1-\delta$ over *i.i.d.* samples S of size at least $m(\epsilon, \delta)$ from P, we have $\mathcal{A}(S) \leq \mathrm{Err}_P(H) + \epsilon$. We call the smallest[1] function that satisfies the above condition the *sample complexity* of the algorithm $\mathcal{A}$ for learning H with respect to $\mathcal{Q}$ and denote it by $m[\mathcal{A}, H, \mathcal{Q}]$. If $H = \{0,1\}^{\mathcal{X}}$ is the class of all functions from $\mathcal{X}$ to $\{0,1\}$, we omit the class in this notation. Thus, we let $m[\mathcal{A}, \mathcal{Q}]$ denote the sample complexity of algorithm $\mathcal{A}$ of *unrestricted learning* with respect to $\mathcal{Q}$.

In the common model of (agnostic PAC) learning a hypothesis class, learnability is defined with respect to the set $\mathcal{Q}$ of *all* distributions over $\mathcal{X} \times \{0,1\}$. However, the lower bounds on the sample complexity of active learning mentioned above show that in the worst case over all data distributions the label complexity of active learning is on par with that of passive learning. We therefore consider learning with respect to restricted sets of data generating distributions.

Active Learning An *active learner* receives an unlabeled sample $S_{\mathcal{X}} = (x_1, \ldots, x_m)$ generated *i.i.d.* by $P_{\mathcal{X}}$ (throughout this paper, we use the notation $S_{\mathcal{X}}$ as opposed to S, if we want to stress that a sample consists of *unlabeled* domain points). The active learner can then sequentially query labels for points in $S_{\mathcal{X}}$, i.e. the learner chooses indices $i_1, \ldots, i_n \in \{1, \ldots m\}$ and receives the labels $l(x_{i_1}), \ldots, l(x_{i_n})$. At each step, the choice of each i_j can depend on $S_{\mathcal{X}}$ and the labels seen so far. Based on the unlabeled sample $S_{\mathcal{X}}$ and the queried labels, the learner outputs a hypothesis.

We say that an algorithm $\mathcal{A}$ *actively learns* some hypothesis class H over $\mathcal{X}$ with respect to a set of distribution $\mathcal{Q}$ over $\mathcal{X} \times \{0,1\}$, if there exist functions $m_u : (0,1) \times (0,1) \to \mathbb{N}$, $m_l : (0,1) \times (0,1) \to \mathbb{N}$, such that, for all $\epsilon, \delta \in (0,1)$, for all distributions $P \in \mathcal{Q}$, with probability at least $1-\delta$ over an *i.i.d.* unlabeled $P_{\mathcal{X}}$-generated sample $S_{\mathcal{X}}$ of size $m_u(\epsilon, \delta)$, the algorithm $\mathcal{A}$ queries at most $m_l(\epsilon, \delta)$ members of S for their labels and $\mathrm{Err}_P(\mathcal{A}(S_{\mathcal{X}})) \leq \mathrm{Err}_P(H) + \epsilon$. Given a function m_u for the size of the unlabeled sample, we say that $\mathcal{A}$ has *labeled sample complexity* or *label complexity* m_l *with respect to* m_u for the smallest function m_l such that the pair of functions (m_l, m_u) that satisfies the above condition. The minimum such function m_l for which there exists a function m_u such that the pair of functions (m_l, m_u) satisfies the above condition is called the *labeled sample complexity* of $\mathcal{A}$ *for actively learning* H with respect to $\mathcal{Q}$ and denoted by $m^{\mathrm{act}}[\mathcal{A}, H, \mathcal{Q}]$. We define $m^{\mathrm{act}}[\mathcal{A}, \mathcal{Q}]$ analogously to the passive counterpart above.

We investigate the sample complexity as a function of $1/\epsilon$. Whenever we use Landau-notation to denote some function growth behavior, this function is considered as a function ϵ only (we consider the asymptotic behavior as ϵ tends to 0), and we omit log-factors.

Probabilistic Lipschitzness We analyze the label complexity of active learning with respect to distribution that have bounded *Probabilistic Lipschitzness (PL)*. PL can be viewed as a way of formalizing the *cluster assumption* of the data, an assumption that is often made to account for the success of semi-supervised learning. It implies that the data can be divided into clusters that are almost label-homogeneous and are separated by low-density regions.

1. Note that the pointwise minimum function over all functions that satisfy the condition, also satisfies the condition. Thus, the "smallest" function is well defined in this context.

Definition 1 (Probabilistic Lipschitzness) *Let $\phi : \mathbb{R} \to [0,1]$. We say that $f : \mathcal{X} \to \mathbb{R}$ is ϕ-Probabilistic Lipschitz with respect to a distribution $P_{\mathcal{X}}$ over $\mathcal{X}$ if for all $\lambda > 0$:*

$$\Pr_{x \sim P_{\mathcal{X}}} \left[\Pr_{y \sim P_{\mathcal{X}}} [\, |f(x) - f(y)| \;>\; (1/\lambda)\, \|x - y\| \,] \;>\; 0 \right] \;\leq\; \phi(\lambda)$$

If, for some $P = (P_{\mathcal{X}}, l)$, the labeling function l is ϕ-Lipschitz, then we say P satisfies the ϕ-Probabilistic Lipschitzness. We denote the set of all such distributions over $[0,1]^d$ by $\mathcal{Q}^d_\phi$. Given some PL-function ϕ and some ϵ, we let $\phi^{-1}(\epsilon)$ denote the smallest λ, such that $\phi(\lambda) \geq \epsilon$.

If a distribution P is ϕ-Lipschitz for some function ϕ, then there always exists a non-decreasing function $\phi' \leq \phi$ (pointwise) such that P is also ϕ'-Lipschitz. We will thus implicitly assume that ϕ is non-decreasing for all PL-functions ϕ considered in this work.

If a distribution $P = (P_{\mathcal{X}}, l)$ is ϕ-Lipschitz, then the weight of points x that have a positive mass of points of opposite label in an λ-ball around them, is bounded by $\phi(\lambda)$. This definition relaxes the standard definition of Lipschitzness. Namely, for points x and y at distance smaller than λ with opposite labels, the standard Lipschitz condition for Lipschitz constant $1/\lambda$ is violated as $|l(x) - l(y)| = 1 > 1/\lambda \|x - y\|$. Thus, if the labeling function l of a distribution is L-Lipschitz (on the support of the distribution) then it satisfies Probabilistic Lipschitzness with the function $\phi(\lambda) = 1$ if $\lambda \geq 1/L$ and $\phi(\lambda) = 0$ if $\lambda < 1/L$. See Steinwart and Christmann (2008) or Urner et al. (2011b) for examples of PL distributions.

4. The PLAL Labeling Procedure

The general framework for our algorithm was suggested in Dasgupta and Hsu (2008). The idea is to use a hierarchical clustering (cluster tree) of the unlabeled data, check the clusters for label homogeneity by starting at the root of the tree (the whole data-set) and working towards the leaves (single data points). The label homogeneity of a cluster is estimated by choosing data points for label queries uniformly at random from the cluster. If a cluster can be considered label homogeneous with sufficiently high confidence, all remaining unlabeled points in the cluster are labeled with the majority label and no further points from this cluster will be queried. If a cluster is detected to be label heterogeneous, it is split into its children in the cluster tree. Since the cluster tree is fixed before any labels were seen, the algorithm can reuse labels from the parent cluster (the induced subsample can be considered a sample that was chosen uniformly at random from the points in the child-cluster) without introducing any sampling bias. Dasgupta (2011) provides nice overview on this.

Dasgupta and Hsu (2008) analyze of this framework assuming that there exist a label homogeneous clustering of the data consisting of a relatively small number of tree-node clusters. In contrast, our analysis depends on the rate in which the diameters of the clusters shrink. Invoking the PL assumption, we can turn such cluster-diameter bounds into error bounds and label query bounds of the procedure. The rates in which cluster diameters shrink have been analyzed for cluster trees that are induced by *spatial trees* in Verma et al. (2012). In this work, we consider a version of the general framework that employs spatial trees for the hierarchical clustering. To obtain a concrete algorithm from the general framework, we also need to specify, how many points to query per cluster and in which order to choose the clusters. We describe our version of this labeling procedure in the next subsection.

Algorithm 1 PLAL labeling procedure

Input: unlabeled sample $S_{\mathcal{X}} = (x_1, \ldots, x_m)$, spatial tree T, parameters ϵ, δ
level $= 0$
active_cells[0].*append*(Root(T))
while *active_cells*[level] not empty **do**
  $q_{\text{level}} = \frac{\text{level} \cdot 2 \cdot \ln(2) + \ln(1/\delta)}{\epsilon}$
  for all C **in** *active_cells*[level] **do**
    C.*query*(q_{level})
    if all labels seen in C are the same **then**
      label all points in $C \cap S$ with that label *(cell C now becomes inactive)*
    else
      if there are unqueried points in $C \cap S$ **then**
        active_cells[level + 1].*append*(Right(C), Left(C))
      end if
    end if
  end for
  level = level + 1
end while
Return: labeled sample $S = ((x_1, y_1) \ldots, (x_m, y_m))$

4.1. The algorithm

A spatial tree is a binary tree T, where each node consists of a subset of the space $\mathcal{X} = [0,1]^d$. We refer to these subsets as *cells*. The root Root(T) of a spatial tree is the whole space $[0,1]^d$ and for each node (cell) C the children Left(C) and Right(C) form a 2-partition of the node C. This implies that for each *level* k (distance from the root), the nodes at this level form a 2^k-partition of the space. For a sample S, a spatial tree induces a hierarchical clustering of S with *clusters* $S \cap C$ for the nodes C in the tree.

Our algorithm works in rounds (see pseudocode in Algorithm 1). It takes an unlabeled *i.i.d.* sample $S_{\mathcal{X}}$ and a spatial tree T as input. At each round, the algorithm maintains a partition of the space $[0,1]^d$ into *active* and *inactive cells*. Initially, there is only one active cell, which is the root of the tree T, i.e. the entire unit cube $[0,1]^d$ containing all sample points. Per round (level), the algorithm queries sufficiently many labels from the $S_{\mathcal{X}}$ points in each of the active cells, to detect if the cell is label heterogeneous (the next paragraph gives a more detailed explanation for this method C.*query*()). A *label homogenous* cell (all seen labels in the cell are the same) is declared inactive and all remaining sample points in the cell are assigned that label. For a *label heterogeneous* cell, the children of the cell in T are added to the list of active cells for the next round, if they still contain unlabeled points.

For a cell C, method C.*query*(q) queries the labels of the first q sample points in the cell. For this, it reuses labels of points that were queried in earlier rounds (i.e. does not actually query those). If the cell contains fewer than q sample points, the labels of all unlabeled points among these are queried and the cell is declared inactive. In this case, it is not important whether the cell is label homogeneous or label heterogeneous, as the algorithm does not infer labels for any of the points and thus all the labels of points in such cells are

correct labels. Note that "declaring a cell inactive" is implicit in the code of Algorithm 1: Only for cells that are heterogeneous *and* contain unlabeled points the children are added to the list of active cells for the next round.

At the end of the procedure all sample points in $S_{\mathcal{X}}$ are labeled. Each point was either queried or obtained an induced label from the homogeneous declared cell it resides in. Only in the latter case, a point might possibly have obtained an erroneous label. We show in Subsection 4.2 below that, by choosing the *query numbers* $q_i = \frac{i \cdot 2 \cdot \ln(2) + \ln(1/\delta)}{\epsilon}$, we can bound the number of labeling mistakes this algorithm makes.

4.2. Error-bound

In this section, we prove that with high probability over the unlabeled input sample, PLAL will label almost all points in the sample correctly. More precisely, we show the following:

Theorem 2 *Let $\mathcal{X} = [0,1]^d$ be the domain, $P_{\mathcal{X}}$ a distribution over $\mathcal{X}$, $l : \mathcal{X} \to \{0,1\}$ a labeling function and $m \in \mathbb{N}$. Then, when given an i.i.d. unlabeled $P_{\mathcal{X}}$-sample $S_{\mathcal{X}}$ of size m and parameters ϵ and δ, with probability at least $(1-\delta)$ (over the choice of the sample $S_{\mathcal{X}}$), PLAL labels at least $(1-\epsilon)m$ many points from $S_{\mathcal{X}}$ correctly.*

Proof Consider a cell that is declared inactive by the PLAL procedure. This cell was either declared homogeneous or all the points in the cell were actually queried for their label. In the latter case, all points receive the correct label. We show that in each cell C, that was declared homogeneous, at most an ϵ-fraction of the points are labeled incorrectly. Note that, at each stage, the set of labeled points in a cell C can be viewed as a superset of a set chosen uniformly at random from the cell: Such a uniformly chosen set may have revealed fewer than q labels, and as $S_{\mathcal{X}}$ is an *i.i.d.* sample we can without loss of generality assume that these are the points with smallest indices in the cell.

Standard analysis shows that, for any cell C, if $\min\{\Pr[l=1|C], \Pr[l=0|C]\} \geq \epsilon$ then a sample of size $\frac{\ln(2/\delta)}{\epsilon}$ has probability at most δ of being label homogeneous. Therefore, choosing query numbers $\frac{\ln(2/\delta_C)}{\epsilon}$, for every cell C, guarantees that with probability at least $1-\delta_C$, it will either be declared homogeneous, resulting in at most an ϵ-fraction of the sample points in the cell being misclassified or the cell will be declared heterogeneous and split further. By choosing $\delta_C = \delta/2^{2k-1}$, where k is the level of the cell C, we ensure that the sum over all confidence parameters δ_C for all cells C, that are declared homogeneous, is at most δ (note this results in our query numbers $\frac{\ln(2/\delta_C)}{\epsilon} = \frac{k \cdot 2 \cdot \ln(2) + \ln(1/\delta)}{\epsilon}$) Thereby, with probability $1-\delta$ over samples, PLAL labels at least a $(1-\epsilon)$-fraction of the points correctly. ■

Remark 3 *It is interesting to note that, if the spatial tree was fixed* before *the unlabeled sample $S_{\mathcal{X}}$ was drawn, then for a given cell C the set of points whose labels were queried can be viewed as a sample from the underlying distribution restricted to this cell. This implies that, when PLAL declares the sample in a cell label homogeneous (after querying the labels of the first $\frac{\ln(2/\delta_C)}{\epsilon}$ sample points in the cell), we can actually conclude that at most an ϵ-fraction (according to the distribution) of all domain points in the cell are of the opposite*

label. Thus, if we restrict our view to the cells that get declared homogeneous during a run of PLAL, the labeling that labels those cells with the detected label has error at most ϵ.

4.3. Bound on the number of queries

We now provide a bound on the number of queries the algorithm makes when fed with an unlabeled sample of size m under the assumption that the data generating distribution satisfies a Probabilistic Lipschitz condition. Our bounds involve the spread of the sample points at level k, called the *data diameter*. In order to avoid overloaded notation, we consider the spatial tree T fixed for this section. For a set of points S, we let λ_k^S denote the maximum data-diameter in a cell at level k, i.e. $\lambda_k^S = \max\{\text{diam}(C,S) \ : \ C \text{ is a cell at level } k\}$, where $\text{diam}(C,S)$ is the *data-diameter* of the sample points in cell C, defined as $\text{diam}(C,S) = \max_{x,y \in C \cap S} \|x-y\|$. The diameter of a cell is always an upper bound on its data-diameter.

Theorem 4 *Let $\mathcal{X} = [0,1]^d$ be the domain, $P_{\mathcal{X}}$ a distribution over $\mathcal{X}$, $l : \mathcal{X} \to \{0,1\}$ a labeling function that is ϕ-Lipschitz for some function ϕ, let $q_i = \frac{i \cdot 2 \cdot \ln(2) + \ln(1/\delta)}{\epsilon}$ denote the query numbers of PLAL for level i and let $(\lambda_i)_{i \in \mathbb{N}}$ be a decreasing sequence with $\lambda_i \in [0, \sqrt{d}]$. Then the expected number of queries that PLAL makes on an unlabeled i.i.d. sample S from $P_{\mathcal{X}}$ of size m, given that the data diameter of S at level k satisfies $\lambda_k^S \leq \lambda_k$ for all k, is bounded by* $\min_{k \in \mathbb{N}} \ (q_k 2^k + \phi(\lambda_k) \cdot m)$.

Proof For each level, the Probabilistic Lipschitzness allows us to bound the number of points that lie in heterogeneous cells at level k: For any sample point x that lies in a label heterogeneous cluster at level k, there is a sample point y in this cluster, such that the labeling function l on x and y violates the (standard) Lipschitz condition for $1/\lambda_k^S$, and thus also for $1/\lambda_k$. The total weight of such points x is bounded by $\phi(\lambda_k)$. Therefore (as λ_k was fixed before drawing the sample), the expected number of sample points that lie in heterogeneous clusters at level k is bounded by $\phi(\lambda_k) \cdot m$. Thus, the expected number of points that are still unlabeled at the beginning of round $k+1$ is bounded by $\phi(\lambda_k) \cdot m$.

Consider the partition of the space PLAL has produced at the beginning of round k (some of the cells in this partition are homogeneous cells from previous rounds and some are the active cells at this level k). Clearly, q_k is a bound on the number of label-queries the algorithm made so far for each of the cells in this partition, as we reuse labels from previous rounds, and the sequence $(q_i)_{i \in \mathbb{N}}$ is non-decreasing. There are at most 2^k cells in this partition. Thus $q_k 2^k$ is an upper bound on the number of queries made up to level k. These two bounds together imply that the number of queries is bounded by $q_k 2^k + \phi(\lambda_k) \cdot m$ for any k. ■

The following corollary will allow us to obtain concrete bounds on the number of queries for various probabilistic Lipschitz functions (see Table 1 below). It follows directly from Theorem 4. Note that, provided the sequence $(q_i)_{i \in \mathbb{N}}$ of query numbers is non-decreasing, the condition $\phi(\lambda_{k^*}) \cdot m \leq q_{k^*} \cdot 2^{k^* d}$ in the corollary is satisfied for sufficiently large k^*: $\phi(\lambda)$ is decreasing for $\lambda \to 0$, and $\lambda_k \to 0$ for $k \to \infty$ (see comment after Definition 1).

Corollary 5 *Under the conditions of Theorem 4, let k^* be such that $\phi(\lambda_{k^*}) \cdot m \leq q_{k^*} \cdot 2^{k^*}$. Then the expected number of queries that PLAL makes on an unlabeled i.i.d. sample from $P_{\mathcal{X}}$ of size m is bounded by* $\frac{k^* \cdot 2 \cdot \ln(2) + \ln(1/\delta)}{\epsilon} \cdot 2^{k^*+1}$.

4.4. Bounds for specific trees

Dyadic trees Here we provide concrete bounds on the expected number of queries for *dyadic trees*. In a dyadic spatial tree, cells are always partitioned by halving one of the coordinates, cycling through the dimensions. That is, for any k, the initial unit cube $[0,1]^d$ (at the root of the tree) is split into 2^{kd} cubes of sidelength $1/2^k$ at level $k \cdot d$. The diameter of such a cube at level kd is $\lambda_{kd} = \sqrt{d}/2^k$, which is at the same time an upper bound on the data diameter λ^S_{kd} at level kd for any sample S.

Table 1 provides an overview on the bounds that we get from Corollary 5 for the polynomial and the exponential Lipschitz assumption. For each of the considered probabilistic Lipschitz functions, we first calculate a value k^* such that $\phi(\lambda_{k^*}) \cdot m \leq q_{k^*} \cdot 2^{k^*}$ and then plug this into the formula of Corollary 5 in order to bound the expected number of queries. The calculations can be found in the appendix.

Table 1: Dyadic trees

Lipschitzness	Bound on expected number of queries
$\phi(\lambda) = \lambda^n$	$2 \cdot \frac{\log(\sqrt{d}^n m\epsilon)^{\frac{d}{n+d}} \ln(2)+\ln(1/\delta)}{\epsilon} \cdot (\sqrt{d}^n m\epsilon)^{\frac{d}{n+d}} = O(m^{\frac{d}{n+d}} \left(\frac{1}{\epsilon}\right)^{\frac{n}{n+d}})$
$\phi(\lambda) = \mathrm{e}^{\frac{-1}{\lambda}}$	$\frac{\sqrt{d}^d \log(\epsilon m)^d}{\epsilon} 2(\log(\log((\epsilon m)^{\sqrt{d}}))d\ln(2) + \ln(2/\delta)) = O(\frac{1}{\epsilon})$

Other spatial trees Often, the *intrinsic dimension* of real data is considerably smaller than the Euclidean dimension of its feature space. Verma et al. (2012) show (for several notions of intrinsic dimension) that, for various classes of spatial trees, the expected data diameter decreases as a function of this intrinsic dimension. Thus, we expect that the query bounds of PLAL used with these trees scale well with the intrinsic dimension.

5. Using PLAL for Active Learning

In this section, we argue that using PLAL with dyadic trees as a pre-procedure can reduce the label complexity of a passive learner. In Section 5.1, we first show that Empirical Risk Minimizers (ERM algorithms) and Regularized Loss Minimizers (RLM algorithms) are robust to the label errors that PLAL might introduce. This implies that for these types of algorithms it is *safe* to use PLAL for labeling, in the sense that it will not increase the error of the learned classifier by much (and using PLAL can never increase the number of labels used). Generalizing this, we then argue that it is safe to use labels from PLAL to mimic the oracle for any statistical learning algorithm. In a second step in Section 5.2, we prove that there are scenarios, where employing PLAL reduces the label complexity of a learning task.

5.1. Robustness of algorithms

In the previous section we have shown how, given any sample, $S = ((x_1, y_1), \ldots, (x_m, y_m))$, the PLAL labeling procedure takes its unlabeled projection $S_{\mathcal{X}}(x_1, \ldots, x_m)$ as input, queries some of the labels and outputs a labeled sample $S' = ((x_1, y'_1), \ldots, (x_m, y'_m))$ such that, with high probability, the number of label errors $|\{i : y_i \neq y'_i\}|$ is bounded (as a function of the Probabilistic Lipschitzness and the number of labels PLAL queried). We show that in many cases such a sample S' suffices for successful learning.

Definition 6 *Given a sequence of labeled instances, $S = ((x_1, y_1), \ldots, (x_m, y_m))$ and $\epsilon \geq 0$, define the ϵ-neighborhood of S as $\mathcal{N}_\epsilon(S) = \{S' = ((x_1, y'_1), \ldots, (x_m, y'_m)) : |\{i : y_i \neq y'_i\}|/m \leq \epsilon\}$. We say that a learning algorithm, $\mathcal{A}$, is $(m, \epsilon, \delta, \eta)$- robust with respect to a data distribution P, if, $\Pr_{S\sim P^m}[\ \forall S' \in \mathcal{N}_\epsilon(S),\ \mathrm{Err}_P(\mathcal{A}(S')) \leq \mathrm{Err}_P(\mathcal{A}(S)) + \eta] \geq (1-\delta)$.*

The next lemma (that upper bounds the error introduced by the use of PLAL for robust algorithms) follows directly from this definition and Theorem 2.

Lemma 7 *Let $\mathcal{A}$ be a learner that is $(m, \epsilon, \delta, \eta)$-robust with respect to a distribution P. Then on random training samples of size m generated by P, replacing the fully labeled sample with one actively labeled by the PLAL (with parameters ϵ, δ), results in deterioration of the error of $\mathcal{A}(S)$ by at most η (with probability greater than $(1-2\delta)$ over the samples).*

Next we show that many common learning algorithms are indeed robust with respect to any data generating distribution, for sufficiently large sample sizes (we explicitly discuss these sizes in the next section). Applying Lemma 7, we then conclude that for such algorithms PLAL can be applied as a preliminary procedure, and reduce the label complexity of learning, in cases where the query numbers required by PLAL are sufficiently small (so that it compensates for the η loss of accuracy). We require the following basic notions:

Definition 8 *We say that a labeled sample S is* ϵ-representative *of H with respect to a data-generating distribution P, if for every $h \in H$, $|\mathrm{Err}_S(h) - \mathrm{Err}_P(h)| \leq \epsilon$. We say that a class H satisfies the* uniform convergence property *with rate $m_H^{UC} : (0,1) \times (0,1) \to \mathbb{N}$ if, for any data generating distribution, P and any $\epsilon, \delta > 0$, for every $m \geq m_H^{UC}(\epsilon, \delta)$, $\Pr_{S\sim P^m}[S$ is ϵ-representative for H with respect to $P] \geq 1-\delta$.*

It is well-known that every class H of finite VC-dimension satisfies the uniform convergence property and that there exists a constant C such that, for every such H we have $m_H^{UC}(\epsilon, \delta) = C\frac{\mathrm{VC}(H)+\log(1/\delta)}{\epsilon^2}$. Recall that an algorithm $\mathcal{A}$ is an Empirical Risk Minimizer for a class H if $\mathcal{A}(S) \in \mathrm{argmin}_{h\in H}\mathrm{Err}_S(h)$. A Regularized Loss Minimizer, $\mathcal{B}$, minimizes a combination of the empirical error and some regularization function $\varphi : H \to \mathbb{R}$, that is $\mathcal{B}(S) \in \mathrm{argmin}_{h\in H}(\mathrm{Err}_S(h) + \varphi(h))$. The following lemma is a straightforward consequence of the above definitions. A full formal proof can be found in the appendix.

Lemma 9 *If $m \geq m_H^{UC}(\epsilon, \delta)$ and $\mathcal{A}$ is an ERM (or RLM) algorithm for H, then $\mathcal{A}$, is $(m, \epsilon, \delta, 4\epsilon)$-robust ($(m, \epsilon, \delta, 6\epsilon)$-robust respectively) with respect to any data distribution P.*

5.1.1. Statistical Algorithms

We now argue that labels from PLAL can also safely be used to mimic the input to *statistical algorithms*. These where first introduced by Ben-David et al. (1990) as "learning by distances" and then by Kearns (1993) who coined the term *statistical query learning* and showed their noise tolerance properties. Following Feldman et al. (2013), we define:

Definition 10 *Let P be a distribution over $\mathcal{X} \times \{0,1\}$. We let STAT_P denote an oracle, which takes as input a function $h : \mathcal{X} \to \{0,1\}$ and a tolerance parameter $\tau > 0$ and returns a value $v \in [\mathrm{Err}_P(h) - \tau, \mathrm{Err}_P(h) + \tau]$. We say that an algorithm is* statistical *if (instead of having direct access to samples from P) it makes calls to STAT_P.*

Statistical algorithms can be implemented in the usual random-sample based learning model by taking a sample S of size $O(1/\tau^2)$ and returning the empirical error of h on S. Many common learning algorithms can be efficiently implemented via statistical algorithms (see e.g., Kearns (1998), Blum et al. (1998)). For our purposes, note that for any $\epsilon > 0$ and $\tau > \epsilon$, queries of the form $\mathrm{STAT}_P(h, \tau)$ can be answered by drawing random unlabeled samples $S_{\mathcal{X}}$ of size $O(1/(\tau - \epsilon)^2)$ and then evaluating h on the output $S' \in \mathcal{N}_\epsilon(S)$ of PLAL.

5.2. Label savings

Table 1 provides an upper bound on the number of label queries the PLAL procedure makes using dyadic trees, given the unlabeled projection of a sample S of size m, to generate a sample $S' \in \mathcal{N}_\epsilon(S)$. We now apply these bounds to show provable reductions in the label complexity achieved by using PLAL as a pre-procedure to passive learning algorithms. Given a passive learning algorithm $\mathcal{A}$, we let $\mathcal{A} \circ \mathrm{PLAL}$ denote the composition of $\mathcal{A}$ with the PLAL procedure. That is, $\mathcal{A} \circ \mathrm{PLAL}$ considers an unlabeled sample $S_{\mathcal{X}}$, applies PLAL to $S_{\mathcal{X}}$ and then applies $\mathcal{A}$ to the resulting labeled sample $S' \in \mathcal{N}_\epsilon(S)$.

Since the PLAL query bounds assume that the data-generating distribution satisfies PL, a fair comparison requires establishing lower bounds for the sample complexity in the passive model (of learning from fully labeled random training samples) under the same PL assumptions. In this section, we consider PL-functions ϕ with $\phi(1) = 1$, in particular the "polynomial PL functions", $\phi(\lambda) = \lambda^n$. In this case, the expected number of queries is bounded by $O(m^{\frac{d}{n+d}} \left(\frac{1}{\epsilon}\right)^{\frac{n}{n+d}})$, see Table 1. For an algorithm with (fully supervised) sample complexity $m = \Theta((1/\epsilon)^\alpha)$, this yields a $O(\left(\frac{1}{\epsilon}\right)^{\frac{n+\alpha d}{n+d}}))$ bound on the expected number of queries. Thus, using PLAL reduces the label complexity whenever $\alpha > 1$.

We start by considering *proper learning*, that is learning a hypothesis class H under the additional requirement that the output classifier is a member of H. Any algorithm that is an ERM or an RLM learner is also a proper learner and we have seen in the previous section that we can use labels from PLAL for these. It is well-known that the sample complexity of proper learning a hypothesis class of finite VC-dimension is lower bounded by $\Omega(1/\epsilon^2)$. This can be readily extended to the case of polynomial PL: Consider two points x and y at distance 1 and let H be such that for every $h \in H$, $h(x) = h(y)$. A distribution in $\mathcal{Q}^d_\phi$ can give two different labels to these points. Then, estimating a bias of $1/2 \pm \epsilon$ on the weight of these two points requires a sample size of $\Omega(1/\epsilon^2)$. However, PLAL would make only two label queries (in this specific situation) and (by the above analysis) have labeled complexity $O(\left(\frac{1}{\epsilon}\right)^{\frac{n+2d}{n+d}}))$ for proper learning of this class with respect to $\mathcal{Q}^d_\phi$ in general.

Theorem 11 *Let $\mathcal{X} = [0,1]^d$, let $d, n, v \in \mathbb{N}$ and let $\phi(\lambda) = \lambda^n$. Then, there is a hypothesis class H of VC-dimension v, such that for any passive proper learner $\mathcal{A}$, $m^{\mathrm{act}}[\mathrm{PLAL} \circ \mathrm{ERM}, H, \mathcal{Q}^d_\phi] = O(\left(\frac{1}{\epsilon}\right)^{\frac{n+2d}{n+d}})$, but $m[\mathcal{A}, H, \mathcal{Q}^d_\phi] = \Omega(\frac{1}{\epsilon^2})$, and thus $m^{\mathrm{act}}[\mathrm{PLAL} \circ \mathrm{ERM}, H, \mathcal{Q}^d_\phi] = o(m[\mathcal{A}, H, \mathcal{Q}^d_\phi])$.*

We now provide upper and lower bounds for unrestricted learning under Probabilistic Lipschitzness. Recall that we defined $\phi^{-1}(\epsilon) = \min\{\lambda \ : \ \phi(\lambda) \geq \epsilon\}$ (see Definition 1).

Theorem 12 *Let $d \in \mathbb{N}$, $d \geq 2$ and let $\phi : \mathbb{R} \to [0,1]$ be some PL-function.*
1.) For every passive learning algorithm $\mathcal{A}$ and every $\epsilon > 0$ there exists a distribution $P \in \mathcal{Q}^d_\phi$ such that, $m < \frac{d}{32\epsilon}(\frac{1}{\phi^{-1}(8\epsilon)})^{d-1}$ implies that $\mathbb{E}_{S\sim P^m}[\mathrm{Err}_P(\mathcal{A}(S))] > \epsilon$, thus $m[\mathcal{A}, \mathcal{Q}^d_\phi] = \Omega(\frac{d}{32\epsilon}(\frac{1}{\phi^{-1}(8\epsilon)})^{d-1})$.
2.) There exists a constant C such that the sample complexity of the 1-Nearest Neighbor algorithm (NN) with respect to the class $\mathcal{Q}^d_\phi$, is $m[\mathrm{NN}, \mathcal{Q}^d_\phi](\epsilon,\delta) \leq C \cdot \frac{2}{\epsilon\delta}\left(\frac{\sqrt{d}}{\phi^{-1}(\epsilon)}\right)^d$.

Proof 1. Lower bound: Recall that standard no-free-lunch results imply that if a learner gets a sample of size less than half of the domain size, then there is a distribution with a deterministic labeling function, such that the expected error of the learner is at least $1/4$.

We construct a distribution on $[0,1]^d$ that satisfies the ϕ-Lipschitzness as follows: We set $P(\bar{0}) = 1 - 8\epsilon$ and distribute the remaining mass of 8ϵ uniformly on points of a grid G of sidelength $\lambda = \phi^{-1}(8\epsilon)$ "at the far side of the surface of" $[0,1]^d$, (i.e. the points $x = (x_1, \ldots x_d)$ where at least one of the x_i has value 1 and the others have values in $\{i\lambda \ : \ 1 \leq i \leq d\}$). Now P is ϕ-Lipschitz under any labeling of these grid points.

There are $|G| \geq d/(\lambda)^{d-1}$ such grid points. We show that with probability at least $1/2$, a sample of size at most m hits less than $|G|/2$ gridpoints. The expected number of such hits is bounded by $8\epsilon m$, formally $\mathbb{E}_{S\sim P^m}[|S \cap G|] = 8\epsilon m$. Now Markov's inequality yields $\Pr_{S\sim P^m}[|S\cap G| > |G|/2] \leq \frac{16\epsilon m}{|G|}$. Now $m < \frac{d}{32\epsilon}(\frac{1}{\phi^{-1}(8\epsilon)})^{d-1}$ and $|G| \geq \frac{d}{(\lambda)^{d-1}} = \frac{d}{(\phi^{-1}(8\epsilon))^{d-1}}$ implies $\Pr_{S\sim P^m}[|S\cap G| > |G|/2] < 1/2$. The above mentioned no-free-lunch result implies that, for any learner $\mathcal{A}$, there is a labeling for the points on G, such that $\mathcal{A}$ has expected error at least $\frac{1}{4}\cdot 8\epsilon = 2\epsilon$ given $|G\cap S| \leq |G|/2$. Since we have shown that this happens with probability at least $1/2$ for samples of size at most m, the learners' expected error over all samples of size at most m is at least ϵ.

2. Upper bound: As in our proof of Theorem 13 in the appendix, we can show that the (ϵ,δ)-sample complexity of 1-NN when the labeling function is deterministic and satisfies (standard) L-Lipschitzness is bounded by $\frac{L^d\sqrt{d}^d}{\epsilon\delta}$. For $\lambda = \phi^{-1}(\epsilon)$ at most an ϵ-fraction of the data does not satisfy the standard $1/\lambda$-Lipschitzness, increasing the error by at most ϵ. ■

Lemma 7 does not imply that Nearest Neighbor is a robust algorithm. In order to show, that using PLAL can also reduce the label complexity of unrestricted learning, we consider a slight variant of the standard 1-NN algorithm and denote this by NN $\circ$ PLAL. Instead of labeling each point by the label of its nearest neighbor in the space, we consider the partition of the space into cells at the end of the run of PLAL, and label each point with the label of its nearest neighbor *within its cell*. If a point falls into a cell that is empty, we label it with the label of its nearest neighbor *within its parent-cell* (note that this one is never empty). This slight modification allows us to show the following:

Theorem 13 *Let $d, n \geq 2$ and let $\phi(\lambda) = \lambda^n$. Applying* PLAL *to the Nearest Neighbor algorithm (in the way described above) results in active sample complexity for learning $\mathcal{Q}_\phi$ that is below the sample complexity of any passive learning algorithm for that class. Namely, for any passive learner $\mathcal{A}$, $m[\mathcal{A}, \mathcal{Q}_\phi] = \Omega((\frac{1}{\epsilon})^{1+\frac{d-1}{n}})$, but $m^{\mathrm{act}}[\mathrm{NN}\circ\mathrm{PLAL}, \mathcal{Q}_\phi] = O((\frac{1}{\epsilon})^{1+\frac{d^2}{n(n+d)}})$, and thus $m^{\mathrm{act}}[\mathrm{NN}\circ\mathrm{PLAL}, \mathcal{Q}_\phi] = o(m[\mathcal{A}, \mathcal{Q}_\phi])$.*

Proof [Sketch] By Remark 3 the error for points in the cells that were declared homogeneous by PLAL is at most ϵ if these points get assigned the label of the box. We now need to bound the error of the NN algorithm in cells, where PLAL ended up querying all labels. Since we can bound the distance between a queried point and the sample point who's label we copy, we apply the Lipschitzness to bound the error. We provide a full formal proof of this in the appendix.

For $\phi(\lambda) = \lambda^n$, the lower bound for unrestricted learning in Theorem 12 becomes $\Omega((\frac{1}{\epsilon})^{1+\frac{d-1}{n}})$. If we apply NN $\circ$ PLAL with samples of size $\Theta((\frac{1}{\epsilon})^{\frac{d+n}{n}})$ (see Theorem 12), we reduced the label complexity to $O((\frac{1}{\epsilon})^{1+\frac{d^2}{n(n+d)}})$ (note that $\frac{d^2}{n(n+d)} \leq \frac{d-1}{n}$ for any $d, n \geq 2$). ■

Next, we analyze using PLAL for learning a hypothesis class of finite VC-dimension.

Theorem 14 *For every $n, v \geq 2$ and $d \geq 3n+1$, there exists a class H over $[0,1]^d$ such that $\mathrm{VC}(H) = v$ and, for every passive learner $\mathcal{A}$, $m^{\mathrm{act}}[\mathrm{PLAL} \circ \mathrm{ERM}, H, \mathcal{Q}_\phi^d] = o(m[\mathcal{A}, H, \mathcal{Q}_\phi^d])$.*

Proof We consider the class $H = \{f_{i,j} : i, j \in \{0,1\}\}$ of functions that are constant on $\mathcal{X} \setminus \bar{0}$. More precisely, we define $f_{i,j}$ to be the function with $f(\bar{0}) = i$ and $f(x) = j$ for $x \neq \bar{0}$. Note, that this is a class of VC-dimension 2. We show that for every $\epsilon < 1/4$ there exist a class of distributions $\mathcal{Q}_\epsilon \subseteq \mathcal{Q}_\phi^d$ such that passively learning the class H with respect to $\mathcal{Q}_\epsilon$ requires a sample size of $\Omega(\frac{1}{\epsilon^{1.5}})$, whereas applying PLAL allows us to learn H with only $O(\frac{1}{\epsilon})$ many queries.

We consider all distributions that have support $\{\bar{0}\} \cup G$, where G is a grid of sidelength $\phi^{-1}(\sqrt{\epsilon})$ such that every point in G has distance at least 1 from $\bar{0}$. As in the proof of Theorem 12, we can construct such a grid with $d\left(\frac{1}{\phi^{-1}(\sqrt{\epsilon})}\right)^{d-1} = d\left(\frac{1}{\epsilon}\right)^{\frac{d-1}{2n}} \geq \left(\frac{1}{\epsilon}\right)^{1.5}$ many points (where the last inequality follows from $d \geq 3n+1$). All distributions in $\mathcal{Q}_\epsilon$ assign weight $1-\sqrt{\epsilon}$ to $\bar{0}$ and distribute the remaining weight $\sqrt{\epsilon}$ uniformly over G. We further allow all labeling functions that assign a $(1/2 - \sqrt{\epsilon})$-fraction of the gridpoints one label (either 0 or 1) and a $(1/2 + \sqrt{\epsilon})$-fraction of the gridpoints the other label (and assign any label to $\bar{0}$). By construction, each of these distributions is ϕ-Lipschitz.

The approximation error of the class H is $\mathrm{Err}_P(H) = \sqrt{\epsilon}(\frac{1}{2} - \sqrt{\epsilon}) = (\frac{\sqrt{\epsilon}}{2} - \epsilon)$ for every distribution $P \in \mathcal{Q}_\epsilon$. Thus, ϵ learning the class H with respect to $\mathcal{Q}_\epsilon$ corresponds to estimating the $\sqrt{\epsilon}$-bias on the grid points. Since $|G| \geq \frac{1}{\epsilon^{1.5}}$, $\Theta\left(\frac{1}{\epsilon}\right)$ many sample points on the grid are necessary and sufficient for estimating this $\sqrt{\epsilon}$-bias (see comment in the appendix). As the total weight of the gridpoints is $\sqrt{\epsilon}$, a random sample from the distribution needs to be of size $\Omega(\frac{1}{\epsilon^{1.5}})$, for that many hits to the grid. However, it is easy to see that PLAL would only query the label of $\bar{0}$ once. Thus, using PLAL for this task, results in label complexity $O(\frac{1}{\epsilon})$. ■

Acknowledgments

This research started while the first and third author were visiting ETH Zurich. We thank Joachim Buhmann and for his support, hospitality and for many insightful discussions. We also thank Andreas Krause for inspiring discussions that helped initiate this work.

References

Maria-Florina Balcan, Andrei Z. Broder, and Tong Zhang. Margin based active learning. In *COLT*, pages 35–50, 2007.

Maria-Florina Balcan, Steve Hanneke, and Jennifer Wortman Vaughan. The true sample complexity of active learning. *Machine Learning*, 80(2-3):111–139, 2010.

Shai Ben-David, Alon Itai, and Eyal Kushilevitz. Learning by distances. In *COLT*, pages 232–245, 1990.

Alina Beygelzimer, Sanjoy Dasgupta, and John Langford. Importance weighted active learning. In *ICML*, page 7, 2009.

Alina Beygelzimer, Daniel Hsu, John Langford, and Tong Zhang. Agnostic active learning without constraints. In *NIPS*, pages 199–207, 2010.

Avrim Blum, Alan M. Frieze, Ravi Kannan, and Santosh Vempala. A polynomial-time algorithm for learning noisy linear threshold functions. *Algorithmica*, 22(1/2):35–52, 1998.

Sanjoy Dasgupta. Analysis of a greedy active learning strategy. In *NIPS*, 2004.

Sanjoy Dasgupta. Coarse sample complexity bounds for active learning. In *NIPS*, 2005.

Sanjoy Dasgupta. Two faces of active learning. *Theor. Comput. Sci.*, 412(19):1767–1781, 2011.

Sanjoy Dasgupta and Daniel Hsu. Hierarchical sampling for active learning. In *ICML*, pages 208–215, 2008.

Sanjoy Dasgupta, Daniel Hsu, and Claire Monteleoni. A general agnostic active learning algorithm. In *ISAIM*, 2008.

Vitaly Feldman, Elena Grigorescu, Lev Reyzin, Santosh Vempala, and Ying Xiao. Statistical algorithms and a lower bound for detecting planted cliques. *STOC 2013*, to appear, 2013.

Alon Gonen, Sivan Sabato, and Shai Shalev-Shwartz. Active learning halfspaces under margin assumptions. *CoRR*, abs/1112.1556, 2011.

Steve Hanneke. A bound on the label complexity of agnostic active learning. In *ICML*, pages 353–360, 2007.

Steve Hanneke. Activized learning: Transforming passive to active with improved label complexity. *Journal of Machine Learning Research (JMLR)*, 13(May):14691587, 2012.

Matti Kääriäinen. Active learning in the non-realizable case. In *ALT*, pages 63–77, 2006.

Michael J. Kearns. Efficient noise-tolerant learning from statistical queries. In *STOC*, pages 392–401, 1993.

Michael J. Kearns. Efficient noise-tolerant learning from statistical queries. *J. ACM*, 45(6): 983–1006, 1998.

Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Springer Publishing Company, Incorporated, 1st edition, 2008. ISBN 0387772413.

Ingo Steinwart and Clint Scovel. Fast rates for support vector machines. 35(2):575–607, 2007.

Ruth Urner, Shai Ben-David, and Shai Shalev-Shwartz. Unlabeled data can speed up prediction time. Supplementay Material, 2011a. URL http://www.cs.uwaterloo.ca/~rurner/SupplementICML2011.pdf.

Ruth Urner, Shai Ben-David, and Shai Shalev-Shwartz. Unlabeled data can speed up prediction time. In *ICML*, 2011b.

Nakul Verma, Samory Kpotufe, and Sanjoy Dasgupta. Which spatial partition trees are adaptive to intrinsic dimension? *CoRR*, abs/1205.2609, 2012.

Appendix

Experiments

We designed experiments on synthetic data to empirically evaluate two aspects of the PLAL labeling framework. The first is how well the PLAL algorithm performs in terms of maintaining low prediction error while reducing the number of required labels. We compare the prediction error induced by a supervised classifier trained on the subset of the data requested by the PLAL, to the error induced by a classifier trained on randomly sampled subset of equal size. The second aspect and perhaps the more interesting one, is how the reduction in labeled sample size relates to the (empirical) probabilistic Lipschitzness of the data. To address this question without assuming access to the true data generating distribution, we use an estimator which adheres to a PL definition in which the probability of finding a λ-close point with different label, is bounded (but is not necessarily zero).

Synthetic Data Description

We generated 3 datasets, each consisting of 2000 samples from a mixture of multivariate Gaussian distributions. The distributions included 4 dense Gaussian, as well as 4 sparse Gaussian, with the same parameters governing the density of each group, but each dataset having different sets of values. We used a different label for the samples associated with each Gaussian, resulting in a multi-label classification task with 8 labels. While the covariance matrix parameters ("dense" and "sparse"), of the each dataset were fixed, we varied the dimensionality of the generated data in the different experiments. We always sampled the dense Gaussian means close to the "corners" of the space, whereas the means of the sparse ones we sampled uniformly at random. This procedure essentially allowed us to create datasets exhibiting different empirical PL behaviors, by varying the covariances of the dense and sparse Gaussian of each dataset. We used diagonal covariance matrices to avoid extra

noise and sampled 80% of the points from the dense Gaussian, and the remaining 20% from the sparse ones. The datasets are denoted by A,B, and C corresponding to: A-0.1 dense variance and 1 sparse variance, B-.01 dense variance and .1 sparse variance, and C-.001 dense variance and .1 sparse variance. With this choice of parameters the datasets can be intuitively casted as the most clusterable being C to the least clusterable, or least separable, being A.

EMPIRICAL PROBABILISTIC LIPSCHITZNESS

We plotted the empirical PL of the datasets A,B, and C with dimensions 5, 15 and 25. The lambda values range between $0 - 10$, for each λ value we calculated the empirical $\phi(\lambda)$ as the percentage of data points having at least λ close neighbor with a different label. The results are shown in Figure 1.

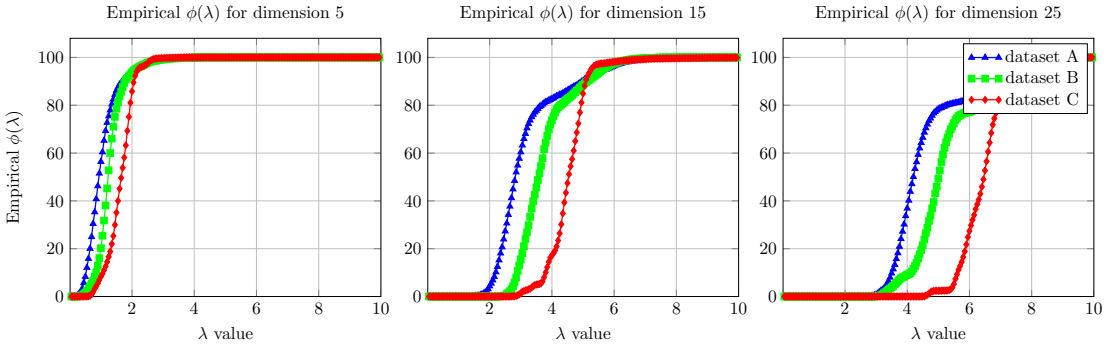

Figure 1: The empirical $\phi(\lambda)$ as a function of λ in the range $0 - 10$ for datasets A,B, and C described in 5.2

CLASSIFICATION

In this experiment we varied ϵ in the range $(0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3)$, and for each ϵ value we computed the PLAL queries. We sampled uniformly at random an equal number of points to serve as a benchmark. We used a K Nearest Neighbor classifier to compute predictions on a test set using the PLAL queries as well as the randomly sampled ones. We used K values in the range $(1, 3, 5, 10)$, and chose the best K for every run. We repeated this procedure 5 times and we report the average values for each configuration. We computed the prediction error as the percentage of labels which differ between the predictions and the true ones. The results on the datasets A,B, and C with dimensions 5, 15 and 25 are shown in Figure 2.

The average number of queries requested by PLAL is plotted in red and is denoted as $\% - \textit{queries}$. The prediction error of the Nearest Neighbor classifier with PLAL queries is denoted as % NN-PLAL-error whereas the prediction error of the Nearest Neighbor classifier with random queries is denoted as % NN-random-error. The plots confirm the intuition that the PLAL labeling framework will save the most labels on datasets which are more clusterable. The empirical PL behavior of the datasets matches the clusterability

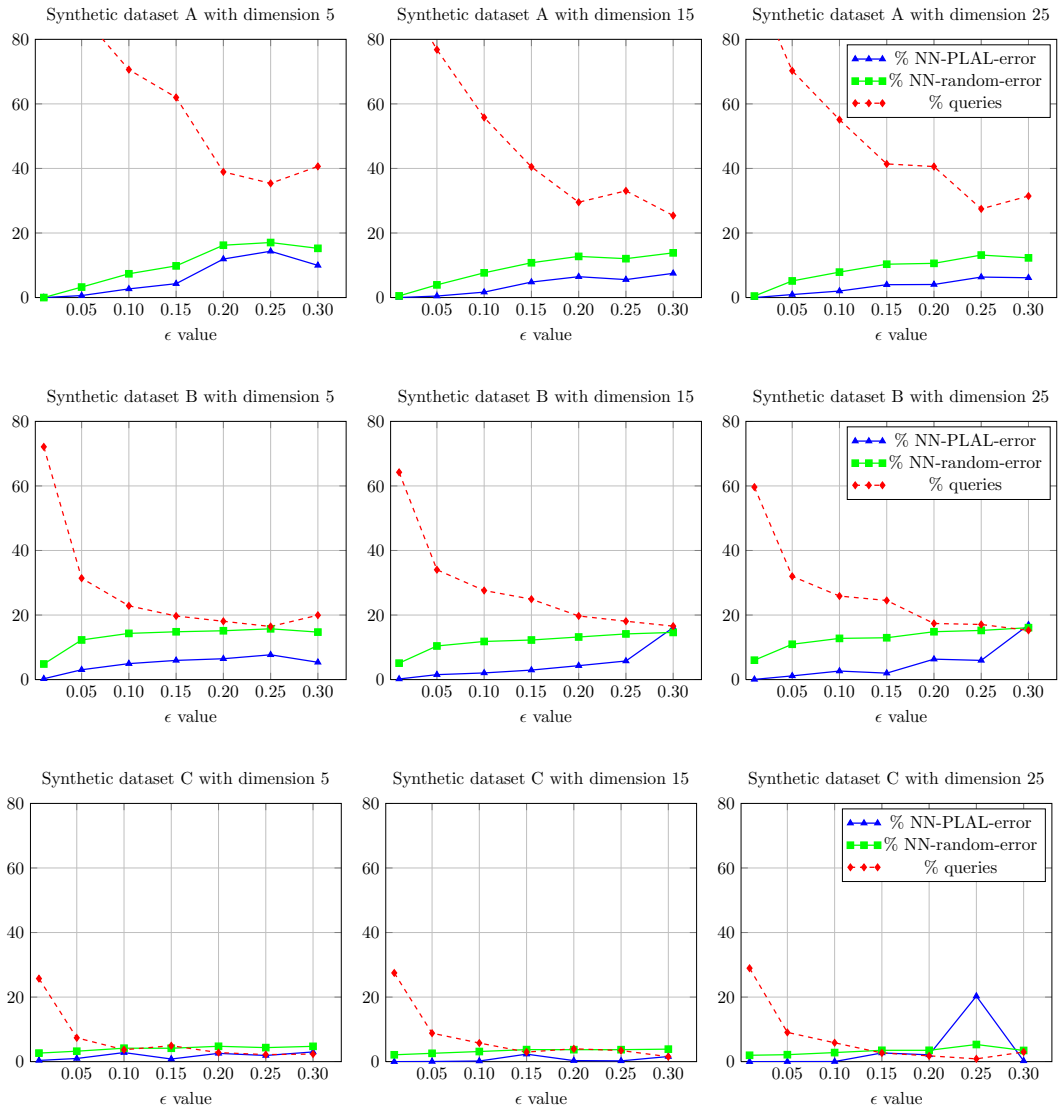

Figure 2: The average number of queries requested by PLAL on datasets A,B, and C for different values of ϵ is denoted as $\%-queries$. The average prediction error of the Nearest Neighbor classifier with PLAL queries is denoted as % NN-PLAL-error and the prediction error of the Nearest Neighbor classifier with random queries is denoted as % NN-random-error

classification. Dataset A for all choices of dimensions, exhibits the fastest increase of $\phi(\lambda)$ whereas dataset C is the slowest. It is evident in the plots that the PLAL algorithm is more sensitive than random sampling to very small sample sizes. On such instances the overall error is close to the bound.

Proofs for Table 1

Polynomial Lipschitzness Assmume $\phi(\lambda) = \lambda^n$. We need to find a k such that

$$\phi(\lambda_k) \cdot m \leq q_k 2^k,$$

where $q_k = \frac{k \cdot 2\ln(2) + \ln(1/\delta)}{\epsilon}$. Note that, if this inequality holds for some some value $k = k^*$ it will also hold for all $k \geq k^*$. We have $\lambda_{kd} = \frac{\sqrt{d}}{2^k}$. We show that for

$$k = \log(\sqrt{d}^n m\epsilon)^{\frac{1}{n+d}}$$

we have

$$\phi(\lambda_{kd}) \cdot m \leq q_{kd} 2^{kd},$$

With the above value for k we get

$$k = \frac{\log(\sqrt{d}^n m\epsilon)}{n+d},$$

thus

$$2^{k(n+d)} = \sqrt{d}^n m\epsilon,$$

thus

$$2^{kd}\frac{1}{\epsilon} \geq \frac{\sqrt{d}^n}{2^{kn}} m,$$

thus

$$2^{kd}\frac{kd \cdot 2\ln(2) + \ln(1/\delta)}{\epsilon} \geq \frac{\sqrt{d}^n}{2^{kn}} m,$$

which is what we needed to show. Thus we can set $k^* = kd = d\log(\sqrt{d}^n m\epsilon)^{\frac{1}{n+d}}$. According to Corrolary 5 the number of queries is now bounded by

$$\begin{aligned} & 2 \cdot \frac{k^* \cdot 2\ln(2) + \ln(1/\delta)}{\epsilon} \cdot 2^{k^*} \\ = \; & 2 \cdot \frac{kd \cdot 2\ln(2) + \ln(1/\delta)}{\epsilon} \cdot 2^{kd} \\ = \; & 2 \cdot \frac{\log(\sqrt{d}^n m\epsilon)^{\frac{d}{n+d}} \ln(2) + \ln(1/\delta)}{\epsilon} \cdot (\sqrt{d}^n m\epsilon)^{\frac{d}{n+d}} = \tilde{O}(m^{\frac{d}{n+d}} (1/\epsilon)^{\frac{n}{n+d}}) \end{aligned}$$

Exponential Lipschitzness Assume $\phi(\lambda) = \mathrm{e}^{-\frac{1}{\lambda}}$. We need to find a k such that

$$\phi(\lambda_k) \cdot m \leq q_k 2^{kd},$$

where $q_k = \frac{k \cdot 2\ln(2) + \ln(1/\delta)}{\epsilon}$. Note that, if this inequality holds for some some value $k = k^*$ it will also hold for all $k \geq k^*$. We have $\lambda_{kd} = \frac{\sqrt{d}}{2^k}$. We show that for

$$k = \log(\log(\epsilon m)^{\sqrt{d}})$$

we have

$$\phi(\lambda_{kd}) \cdot m \leq q_{kd} 2^{kd},$$

With the above value for k we get

$$\frac{2^k}{\sqrt{d}} = \log(\epsilon m),$$

thus

$$kd + \frac{2^k}{\sqrt{d}} \log(e) \geq \log(\epsilon m),$$

thus

$$2^{kd} \mathrm{e}^{\frac{2^k}{\sqrt{d}}} \geq \epsilon m,$$

thus

$$(kd2\ln(2) + \ln(1/\delta)) \cdot 2^{kd} \mathrm{e}^{\frac{2^k}{\sqrt{d}}} \geq \epsilon m,$$

thus

$$\frac{(kd2\ln(2) + \ln(1/\delta))}{\epsilon} \cdot 2^{kd} \geq \mathrm{e}^{\frac{2^k}{\sqrt{d}}} m,$$

which is what we needed to show. Thus we can set $k^* = kd = d\log(\log(\epsilon m)^{\sqrt{d}})$. According to Corrolary 5 the number of queries is now bounded by

$$\begin{aligned}
& 2 \cdot \frac{k^* \cdot 2\ln(2) + \ln(1/\delta)}{\epsilon} \cdot 2^{k^*} \\
= \; & 2 \cdot \frac{kd \cdot 2\ln(2) + \ln(1/\delta)}{\epsilon} \cdot 2^{kd} \\
= \; & \frac{2(\log(\log(\epsilon m)^{\sqrt{d}})d\ln(2) + \ln(1/\delta))}{\epsilon} \cdot (\log(\epsilon m)^{\sqrt{d}})^d \\
= \; & \frac{\sqrt{d}^d \log(\epsilon m)^d}{\epsilon} 2(\log(\log((\epsilon m)^{\sqrt{d}}))d\ln(2) + \ln(1/\delta))
\end{aligned}$$

Proof of Lemma 9

ERM algorithms For some sample S, we let h_S denote the empirical risk minimizer in H with respect to S, i.e. $h_S = \mathrm{argmin}_{h \in H} \mathrm{Err}_S(h)$. By Definition 6 we need to show that

$$\Pr_{S \sim P^m} \left[\forall S' \in \mathcal{N}_\epsilon(S),\ \mathrm{Err}_P(h_{S'})) \leq \mathrm{Err}_P(h_S) + 4\epsilon \right] \geq (1 - \delta).$$

By the Definition 8 (uniform convergence property) we know that a sample of size at least $m \geq m_H^{UC}(\epsilon, \delta)$ is ϵ-representative for H with probability at least $1 - \delta$. Thus, we now assume that the sample S is ϵ-representative and it remains to show that we have for all $S' \in \mathcal{N}(S)$:

$$\mathrm{Err}_P(h_{S'}) \leq \mathrm{Err}_P(h_S) + 4\epsilon.$$

We have

$$\begin{aligned}
\mathrm{Err}_P(h_{S'}) &\leq \mathrm{Err}_S(h_{S'}) + \epsilon && \text{as } S \text{ is } \epsilon\text{-representative} \\
&\leq \mathrm{Err}_{S'}(h_{S'}) + 2\epsilon && \text{as } S' \in \mathcal{N}(S) \\
&\leq \mathrm{Err}_{S'}(h_S) + 2\epsilon && \text{by definition of } h_{S'} \\
&\leq \mathrm{Err}_S(h_S) + 3\epsilon && \text{as } S' \in \mathcal{N}(S) \\
&\leq \mathrm{Err}_P(h_S) + 4\epsilon && \text{as } S \text{ is } \epsilon\text{-representative}
\end{aligned}$$

RLM algorithms Now, for some sample S, we let h_S denote the regularized risk minimizer in H with respect to S, i.e. $h_S = \mathrm{argmin}_{h \in H}(\mathrm{Err}_S(h) + \varphi(h))$. Again, we assume that the sample S is ϵ-representative and now need to show that we have for all $S' \in \mathcal{N}(S)$:

$$\mathrm{Err}_P(h_{S'}) \leq \mathrm{Err}_P(h_S) + 6\epsilon$$

We start by proving that

$$\varphi(h_S) - \varphi(h_{S'}) \leq 2\epsilon \qquad (*)$$

By way of contradiction, let us assume that, on the contrary, $\varphi(h_S) > \varphi(h_{S'}) + 2\epsilon$. Then we get

$$\begin{aligned}
\mathrm{Err}_S(h_S) + \varphi(h_S) &> \mathrm{Err}_S(h_S) + \varphi(h_{S'}) + 2\epsilon \\
&\geq \mathrm{Err}_{S'}(h_S) + \varphi(h_{S'}) + \epsilon && \text{as } S' \in \mathcal{N}(S) \\
&\geq \mathrm{Err}_{S'}(h_{S'}) + \varphi(h_{S'}) + \epsilon && \text{by definition of } h_{S'} \\
&\geq \mathrm{Err}_S(h_{S'}) + \varphi(h_{S'}) && \text{as } S' \in \mathcal{N}(S)
\end{aligned}$$

This contradicts the definition of h_S. With this, we conclude:

$$\begin{aligned}
\mathrm{Err}_P(h_{S'}) &\leq \mathrm{Err}_S(h_{S'}) + \epsilon && \text{as } S \text{ is } \epsilon\text{-representative} \\
&\leq \mathrm{Err}_{S'}(h_{S'}) + 2\epsilon && \text{as } S' \in \mathcal{N}(S) \\
&\leq \mathrm{Err}_{S'}(h_S) + (\varphi(h_S) - \varphi(h_{S'})) + 2\epsilon && \text{by definition of } h_{S'} \\
&\leq \mathrm{Err}_{S'}(h_S) + 4\epsilon && \text{by } (*) \\
&\leq \mathrm{Err}_S(h_S) + 5\epsilon && \text{as } S' \in \mathcal{N}(S) \\
&\leq \mathrm{Err}_P(h_S) + 6\epsilon && \text{as } S \text{ is } \epsilon\text{-representative}
\end{aligned}$$

Proof of Theorem 13

We adapt a proof from Urner et al. (2011b) for the success of the 1-Nearest Neighbor algorithm under Lipschitzness to its modified version of 1-NN with PLAL. We will here prove the following:

Lemma 15 *Let P be a distribution over $[0,1]^d$ with PL-function $\phi(\lambda) = \lambda^n$. Then applying* NN $\circ$ PLAL *to an unlabeled sample $S_{\mathcal{X}}$ of size*

$$m \geq \left(\frac{1}{\epsilon}\right)^{\frac{d}{n}+1} \frac{(2\sqrt{d})^d}{\delta \mathrm{e}}$$

results in classification error at most 2ϵ with probability at least $(1-\delta)$ (over the choice of $S_{\mathcal{X}}$).

For this, we need the following result that also appears in Urner et al. (2011a):

Lemma 16 *Let $C_1, C_2, \ldots, C_r$ be a set of subsets of some domain set $\mathcal{X}$ and let S be a set of points of size m, sampled i.i.d. according to some distribution P over $\mathcal{X}$. Then we have*

$$\mathbb{E}_{S\sim P^m}\left[\sum_{i:C_i\cap S=\emptyset} P[C_i]\right] \leq \frac{r}{m\mathrm{e}}$$

Let $\lambda = \sqrt{d}/2^k$ for the smallest k such that $\sqrt{d}/2^k \leq \phi^{-1}(\epsilon)$. This implies $\phi^{-1}(\epsilon) \geq \lambda \geq \phi^{-1}(\epsilon)/2$. We can cover $\mathcal{X} = [0,1]^d$ with $r = \left(\sqrt{d}/\lambda\right)^d$ boxes $C_1, C_2, \ldots C_r$ of side-length $\lambda/\sqrt{d} = 1/2^k$. Note that any two points inside such a box are at distance at most λ.

Using Markov's inequality, Lemma 16 implies that for any $\epsilon > 0$ and m we have

$$\mathrm{Prob}_{S\sim P^m}\left[\left[\sum_{i:C_i\cap S=\emptyset} P[C_i]\right] > \epsilon\right] \leq \frac{r}{\epsilon m\mathrm{e}}$$

It follows that in this setting, for any $\epsilon, \delta > 0$, a sample of size

$$m \;\geq\; \left(\frac{\sqrt{d}}{\lambda}\right)^d \frac{1}{\epsilon\delta\mathrm{e}} \;=\; \frac{r}{\epsilon\delta\mathrm{e}}$$

suffices to guarantee that with probability exceding $(1-\delta)$, at most an ϵ-fraction of domain points are in boxes that are not hit by the sample. By noting that $\phi^{-1}(\epsilon) = \epsilon^{1/n}$ (for the polynomial Lipschitzness function $\phi(\lambda) = \lambda^n$) and recalling that $\lambda \geq \phi^{-1}(\epsilon)/2$, we obtain that

$$\left(\frac{2\sqrt{d}}{\epsilon^{1/n}}\right)^d \frac{1}{\epsilon\delta\mathrm{e}} \;=\; \left(\frac{2\sqrt{d}}{\phi^{-1}(\epsilon)}\right)^d \frac{1}{\epsilon\delta\mathrm{e}} \;\geq\; \left(\frac{\sqrt{d}}{\lambda}\right)^d \frac{1}{\epsilon\delta\mathrm{e}} \;=\; \frac{r}{\epsilon\delta\mathrm{e}}.$$

Therefore, the sample size stated above suffices for hitting all but an ϵ-fraction of the boxes.

Now consider the modified 1-NN labeling rule, where every point x gets the label of its Nearest Neighbor within the cell that a run of PLAL produced on a samle $S_{\mathcal{X}}$. We denote by S the sample $S_{\mathcal{X}}$ with the labels from PLAL. We refer to the elements of the partition that PLAL produced as *cells* and to the elements of the partition in the argument above as *boxes*. All these elements are axis-alligned rectangles that have powers of $1/2$ as sidelengths. For a point x, we denote the box that contains x by $b(x)$ and the cell that contains x by

$c(x)$. As the sidelengths of both boxes and cells are powers of $1/2$, and we use the dyadic spatial trees, we have $b(x) \subset c(x)$ or $c(x) \subset b(x)$ or $b(x) = c(x)$ for all x.

To bound the probability that a test point x receives the wrong label, we consider the following cases:

Case 1: $c(x)$ was declared homogeneous by PLAL.
Then x will receive the label of $c(x)$. By Remark 3, the total error resulting from such cases is at most ϵ.

Case 2: $c(x)$ was not declared homogeneous by PLAL and $b(x) \subseteq c(x)$.
We chose the sample size of S so that (with probability at least $1-\delta$) at most an ϵ-fraction of points lie in boxes that are not hit by S, thus the probability (over the choice of x) that $S \cap b(x) = \emptyset$ is bounded by ϵ. If $S \cap b(x) \neq \emptyset$, then the Nearest Neighbor of x inside $c(x)$ has distance at most λ from x (recall that the diameter of $b(x)$ is λ). As $\phi^{-1}(\epsilon) \geq \lambda$, at most an ϵ fraction of points x are at distance less than λ from some point of oppostite label. Thus, the error of our labeling rule in this case is at most 2ϵ.

Case 3: $c(x)$ was not declared homogeneous by PLAL, $c(x) \subset b(x)$ and $c(x) \cap S \neq \emptyset$.
We can bound the probability that x receives a wrong label by 2ϵ in the same way as in Case 2. (The probability that $b(x) \cap S = \emptyset$ is bounded by ϵ and otherwise x receives the label of a point that is at distance at most λ.)

Case 4: $c(x)$ was not declared homogeneous by PLAL, $c(x) \subset b(x)$ and $c(x) \cap S = \emptyset$.
In this case x receives the label of its Nearest Neighbor in the parent cell of $c(x)$. We denote this cell by $p(c(x))$. The cell $c(x)$ was produced when PLAL decided to split $p(c(x))$. Thus the parent cell $p(c(x))$ contains points from S. Note that $c(x) \subset b(x)$ implies $p(c(x)) \subseteq b(x)$. This implies that the Nearest Neighbor of x in $p(c(x))$ is at distance at most λ from x and as under Case 2 we bound the probability that this neighbor has a different label than x by ϵ.

Comment on proof of Theorem 14

We use the following lemma in the proof:

Lemma 17 *Let $\mathcal{X}$ be a domain of size at least $1/\epsilon^3$ and let $\mathcal{Q}$ be the set of distributions over $\mathcal{X} \times \{0,1\}$ whose marginal distribution over $\mathcal{X}$ is uniform and whose labeling function deterministically labels a $(1-\epsilon)$-fraction of the points 0 and $(1+\epsilon)$-fraction of the points 1, or the other way around. Let H be the hypothesis class that contains only the constant function 1 and the constant function 0. Then, learning H with respect to $\mathcal{Q}$ with accuracy at most $\epsilon/2$ requires a sample size of $\Omega(1/\epsilon^2)$.*