PLAL: cluster-based active learning

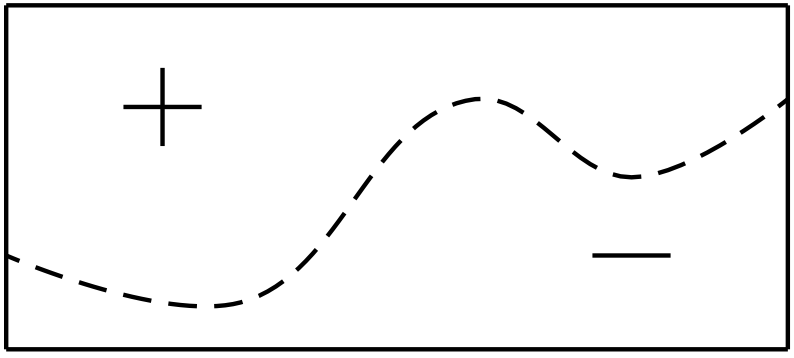
Ruth Urner, Sharon Wulff and Shai Ben-David

COLT 2013, Princeton

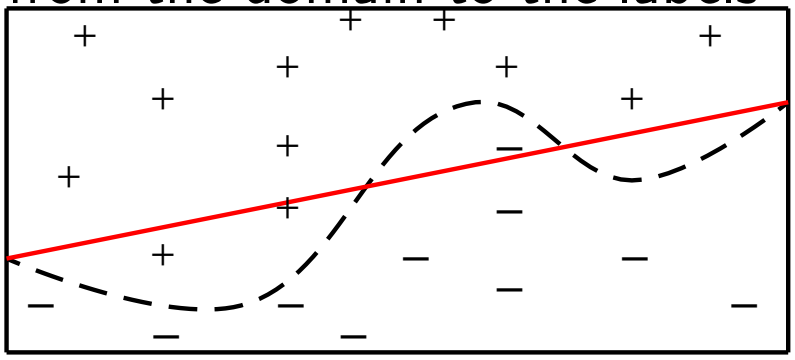
June 13, 2013

Standard Statistical Learning framework

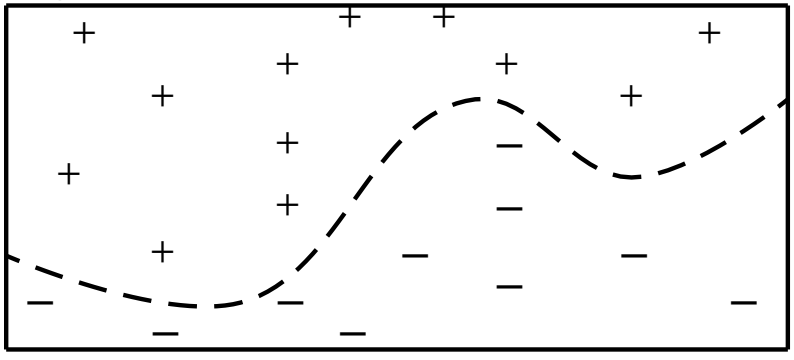
Task: Probability distribution P over a labeled domain space



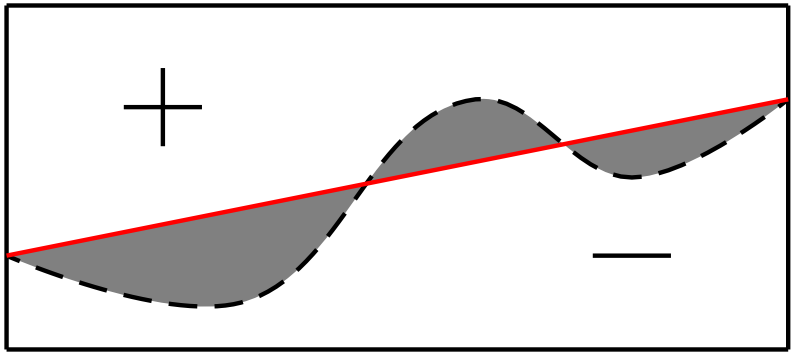
Learner: Produces a function from the domain to the labels



Input: An *i.i.d.* labeled sample from P

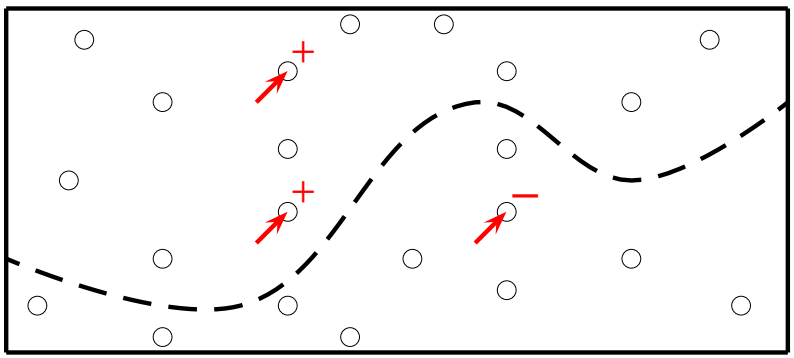


Goal: Minimize the error of the function with respect to P



Active Learning (AL)

Initially, only unlabeled data is available and queries for labels are expensive.



Input:

- unlabeled examples
- labels upon query

The learner aims to build a classifier while making as few label-queries as possible.

Formal model for Active Learning

Domain: $\mathcal{X} = [0,1]^d$

Label set: $\{0,1\}$

Data generating distribution: P over $\mathcal{X} \times \{0,1\}$
generating examples labeled according to $l : \mathcal{X} \to \{0,1\}$
with marginal Distribution: $P_{\mathcal{X}}$ over $\mathcal{X}$

A *classifier* h is a function $h : \mathcal{X} \to \{0,1\}$

Problem:

Input: An unlabeled training sample S *i.i.d.* from $P_{\mathcal{X}}$

Goal: Choose as few as possible points from S to be labeled, and
Learn a classifier h with small error

$$\mathrm{Err}_P(h) := \Pr_{(x,y)\sim P}[h(x) \neq y]$$

Challenges of Active Learning

Sampling bias:

- If the learner chooses which points from an unlabeled sample to label, the resulting set of labeled points may not be a good representation of P (not an *i.i.d.* sample).

Lower bounds/Need for data assumptions:

- Lower bounds for both the realizable and the agnostic case show that, in the worst case, AL requires as many labels as passive learning in general
- Thus, advantages of AL are possible only under additional data assumptions
- Previous analysis of AL is mostly based on the *disagreement coefficient* (Hanneke, 2007)

Previous work on Active Learning

Lower bounds Dasgupta (2005), Kääriäinen (2006), Beygelzimer et al. (2009)

Realizable case/Separability with margin Dasgupta (2004), Balcan et al. (2007), Balcan et al. (2010), Gonen et al. (2012)

Agnostic case Hanneke (2007), Dasgupta (2008), Beygelzimer et al. (2009), Beygelzimer et al. (2010)

Activized Learning Hanneke (2013)

Cluster-based Dasgupta and Hsu (2008)

Previous work on Active Learning

Lower bounds Dasgupta (2005), Kääriäinen (2006), Beygelzimer et al. (2009)

Realizable case/Separability with margin Dasgupta (2004), Balcan et al. (2007), Balcan et al. (2010), Gonen et al. (2012)

Agnostic case Hanneke (2007), Dasgupta (2008), Beygelzimer et al. (2009), Beygelzimer et al. (2010)

Activized Learning Hanneke (2013)

Cluster-based Dasgupta and Hsu (2008)

Previous work on AL: Cluster-based

DH algorithm [Dasgupta and Hsu 2008]

Input: Hierarchical clustering of unlabeled sample $S_{\mathcal{X}}$, and ϵ, δ

Starting with root-cluster, recurse down the tree:

- Choose points uniformly at random from cluster, query labels
- Decide if cluster is label-homogeneous or not
- If heterogeneous, split and recurse on child-clusters
- If homogenous, label all points in cluster with that label

Output: Labeled sample S

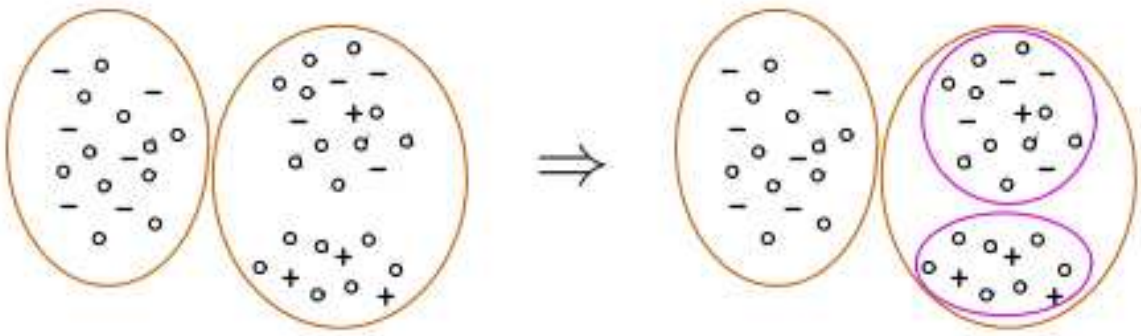


Image from [Dasgupta, 2011]

Previous work on AL: Cluster-based

DH algorithm [Dasgupta and Hsu 2008]

Input: Hierarchical clustering of unlabeled sample $S_{\mathcal{X}}$, and ϵ, δ

Starting with root-cluster, recurse down the tree:

- Choose points uniformly at random from cluster, query labels
- Decide if cluster is label-homogeneous or not
- If heterogeneous, split and recurse on child-clusters
- If homogenous, label all points in cluster with that label

Output: Labeled sample S

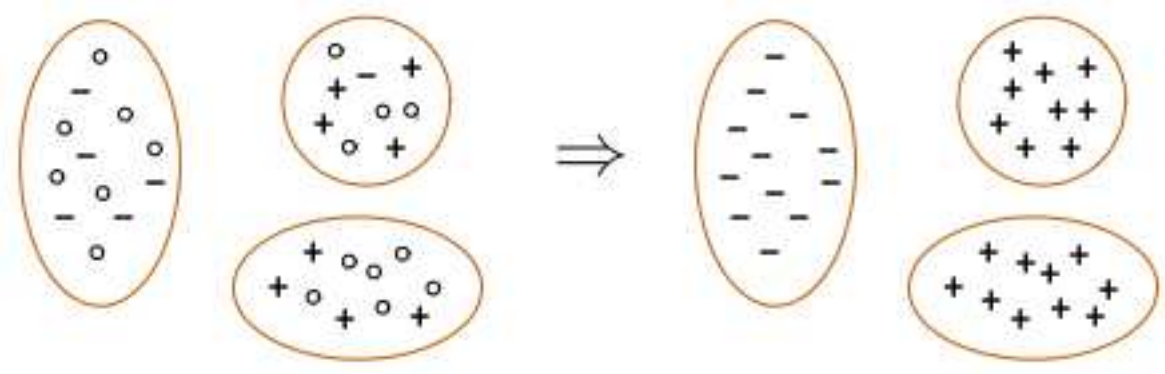


Image from [Dasgupta, 2011]

Previous work on AL: Cluster-based

DH algorithm [Dasgupta and Hsu 2008]

Input: Hierarchical clustering of unlabeled sample $S_{\mathcal{X}}$, and ϵ, δ

Starting with root-cluster, recurse down the tree:

- Choose points uniformly at random from cluster, query labels
- Decide if cluster is label-homogeneous or not
- If heterogeneous, split and recurse on child-clusters
- If homogenous, label all points in cluster with that label

Output: Labeled sample S

Insight: Avoids sampling bias since the cluster split is independent of the labels.

Proposal: Use this labeling paradigm as a pre-procedure to other learning algorithms.

Our Contributions

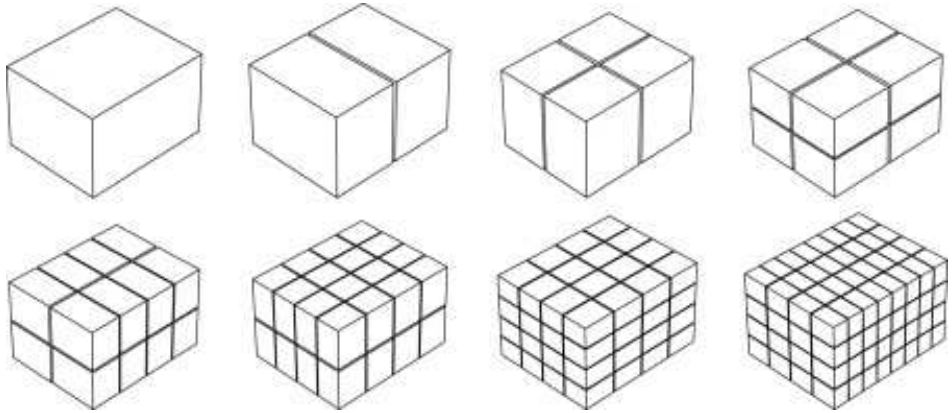
We propose a concrete version (PLAL) of this cluster-based AL paradigm and provide performance guarantees.

We show that the DH paradigm provably saves labels,

- under a general clusterability data-assumption,
- for various types of learning settings and algorithms.

Convert DH framework to an algorithm: PLAL

- Hierarchical clustering: (dyadic) spatial trees



- Choose clusters level by level
- Query $q_k = \frac{k \cdot 2 \cdot \ln(2) + \ln(1/\delta)}{\epsilon}$ many labels in a cluster at level k
- Declare homogeneous if all seen labels are the same, otherwise split

Overview

1. Error bound: PLAL mislabels at most an ϵ-fraction of the sample points.

2. Label-query bound: We bound the number of queries that PLAL makes in terms of Probabilistic Lipschitzness (a notion of clusterability, we will define).

3. We identify learning paradigms, that are robust to the type of label-errors that PLAL introduces.

4. We present several settings, where using PLAL reduces the label complexity.

Overview

1. Error bound: PLAL mislabels at most an ϵ-fraction of the sample points.

2. Label-query bound: We bound the number of queries that PLAL makes in terms of Probabilistic Lipschitzness (a notion of clusterability, we will define).

3. We identify learning paradigms, that are robust to the type of label-errors that PLAL introduces.

4. We present several settings, where using PLAL reduces the label complexity.

Error bound

Theorem

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.

Overview

1. Error bound: PLAL mislabels at most an ϵ-fraction of the sample points.

2. Label-query bound: We bound the number of queries that PLAL makes in terms of Probabilistic Lipschitzness (a notion of clusterability, we will define).

3. We identify learning paradigms, that are robust to the type of label-errors that PLAL introduces.

4. We present several settings, where using PLAL reduces the label complexity.

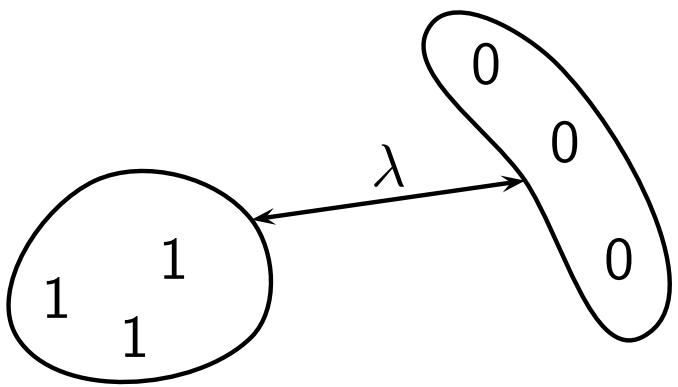
Number of queries depends on clusterability

Intuition: PLAL makes few queries if dense cells are label-homogeneous.

This holds if the class boundaries goes through low-density regions.

Cluster assumption and Lipschitzness

The labeling function satisfies the Lipschitz-condition, only if the data is strongly clusterable:



Lipschitz condition:
$|l(x) - l(y)| \leq 1/\lambda \, \|x - y\|$

The Probabilistic Lipschitzness assumption

Let $\phi : \mathbb{R} \to [0, 1]$. We say that distribution P with labeling function l satisfies the ϕ-Probabilistic Lipschitz (PL) assumption if for all $\lambda > 0$:

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

We assume $\phi(\lambda) = \mathrm{poly}(\lambda)$.

(A version of this notion was introduced by (Steinwart and Scovel, 2005))

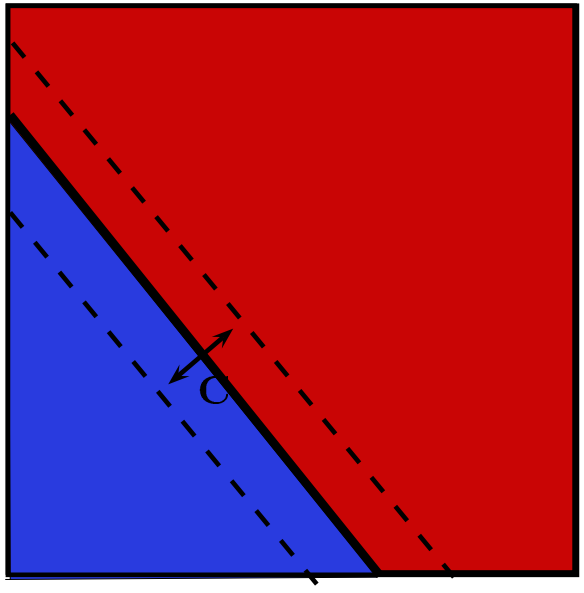
PL examples

Let $P_{\mathcal{X}}$ be the uniform distribution over $\mathcal{X} = [0,1]^d$.

If l is a linear separator then

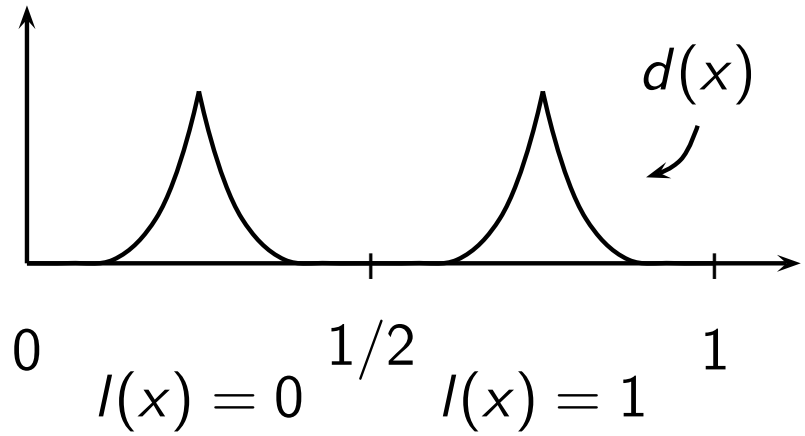
$$\phi(\lambda) = C\lambda$$

for some constant C.



Example–Smoothly clustered data

Domain: $[0, 1]$
Density: forms clusters



Satisfies the new measure of clusterability:

$$\phi(\lambda) = \lambda^n$$

or even

$$\phi(\lambda) = \mathrm{e}^{-1/\lambda}$$

General bound on the number of queries

Theorem

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

$$\min_{k \in \mathbb{N}} \ (q_k 2^k + \phi(\lambda_k) \cdot m).$$

General bound on the number of queries

Proof idea for bound:

$$\min_{k \in \mathbb{N}} \ (q_k 2^k + \phi(\lambda_k) \cdot m).$$

For every k:

Queries for levels up to k: $q_k 2^k$

- q_k - bound on the number of labels queried in each cell in the partition
- 2^k - bound on the number of cells

Queries for levels greater than k: $\phi(\lambda_k^S) \cdot m$

- bounds the expected number of points that lie in heterogeneous cells at level k

Bound for dyadic trees

Lipschitzness	Bound on expected number of queries
$\phi(\lambda) = \lambda^n$	$\tilde{O}(m^{\frac{d}{n+d}} \left(\frac{1}{\epsilon}\right)^{\frac{n}{n+d}})$

$\Rightarrow$ Sample complexity $m = \Theta\left(\frac{1}{\epsilon^\alpha}\right)$ reduced by PLAL whenever $\alpha > 1$.

Overview

1. Error bound: PLAL mislabels at most an ϵ-fraction of the sample points.

2. Label-query bound: We bound the number of queries that PLAL makes in terms of Probabilistic Lipschitzness (a notion of clusterability, we will define).

3. We identify learning paradigms, that are robust to the type of label-errors that PLAL introduces.

4. We present several settings, where using PLAL reduces the label complexity.

Using PLAL as a pre-procedure

We show that we can use PLAL for

- ERM and RLM learners
- Statistical learning algorithms
- Nearest Neighbor learning

We need to show that these learning algorithms are robust to the type of labeling error that PLAL introduces.

Robustness of Algorithms

Definition

Given a sample $S = ((x_1, y_1), \ldots, (x_m, y_m))$ and $\epsilon \geq 0$, define the ϵ-neighborhood of S as

$$\mathcal{N}_\epsilon(S) = \{S' = ((x_1, y_1'), \ldots, (x_m, y_m')) : \ |\{i : y_i \neq y_i'\}|/m \leq \epsilon\}.$$

Definition

We say that a learning algorithm $\mathcal{A}$ is $(m, \epsilon, \delta, \eta)$-robust with respect to a data distribution P, if,
$\Pr_{S \sim P^m}[\ \forall S' \in \mathcal{N}_\epsilon(S),\ \mathrm{Err}_P(\mathcal{A}(S')) \leq \mathrm{Err}_P(\mathcal{A}(S)) + \eta] \geq (1-\delta)$.

$\Rightarrow$ It is *safe* to use PLAL for labeling with *robust* algorithms.

ERM and RLM algorithms are robust

Given a sample S,

- an ERM algorithm outputs:

$$h_S = \text{argmin}_{h \in H} \text{Err}_S(h)$$

- an RLM algorithm outputs

$$h_S = \text{argmin}_{h \in H} (\text{Err}_S(h) + \varphi(h))$$

PLAL labels do not change the empirical error $\text{Err}_S(h)$ by much.

If m is large enough for (ϵ, δ)-uniform convergence of H, then an ERM (or RLM) algorithm is $(m, \epsilon, \delta, 4\epsilon)$-robust ($(m, \epsilon, \delta, 6\epsilon)$-robust respectively).

$\Rightarrow$ It is safe to use labels from PLAL for ERM and RLM learners.

Use PLAL for Statistical Algorithms

We say that an algorithm is *statistical* if (instead of having direct access to samples from P) it makes calls to an oracle that, for query (h, τ) returns a value $v \in [\mathrm{Err}_P(h) - \tau, \mathrm{Err}_P(h) + \tau]$.

This oracle can be realized by returning the empirical error of h.

$\Rightarrow$ It is safe to use labels from PLAL to mimic the oracle for any statistical learning algorithm.

Use PLAL for Nearest Neighbor learning

Modify Nearest Neighbor algorithm:

- Consider the partition of the space into cells at the end of the run of PLAL.
- 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* (this one is never empty).

$\Rightarrow$ For this version of NN, we can use a sample labeled by PLAL.

Can use PLAL with a modified Nearest Neighbor algorithm

Proof idea:

Case 1 A testpoint falls into a cell the was declared homogeneous by PLAL

- The error of assigning the majority label in these areas is at most ϵ.

Case 2 A testpoint falls into a cell, where all sample points were queried

- These points are labeled correctly, thus induce the usual error of NN.

Overview

1. Error bound: PLAL mislabels at most an ϵ-fraction of the sample points.

2. Label-query bound: We bound the number of queries that PLAL makes in terms of Probabilistic Lipschitzness (a notion of clusterability, we will define).

3. We identify learning paradigms, that are robust to the type of label-errors that PLAL introduces.

4. We present several settings, where using PLAL reduces the label complexity.

Reductions in label complexity

We show that using PLAL as a labeling pre-procedure reduces the label complexity of:

1. Proper Learning of a VC-class.
2. Unrestricted Learning of a VC-class.
3. Nearest Neighbor Learning.

For each case, we establish lower bounds for passive learning under Probabilistic Lipschitzness, that are higher than the upper bounds with PLAL.

Reductions in label complexity of learning

Our lower bounds for (passive) learning under Probabilistic Lipschitzness imply the following reductions in labeled sample complexity:

	Passive	PLAL-Active
Proper Learning of H	$\Omega\left(1/\epsilon^2\right)$	$O\left(\left(\frac{1}{\epsilon}\right)^{\frac{n+2d}{n+d}}\right)$
Unrestricted Learning of H	$\Omega\left(\frac{1}{\epsilon^{1.5}}\right)$	$O\left(\frac{1}{\epsilon}\right)$
Nearest Neighbor Learning	$\Omega\left(\left(\frac{1}{\epsilon}\right)^{1+\frac{d-1}{n}}\right)$	$O\left(\left(\frac{1}{\epsilon}\right)^{1+\frac{d^2}{n(n+d)}}\right)$

Summary

We propose a concrete version (PLAL) of the cluster-based AL paradigm and provide performance guarantees.

We show that the DH paradigm provably saves labels,

- under a general clusterability data-assumption,
- for various types of learning settings and algorithms.
  1. Proper Learning of a VC-class.
  2. Unrestricted Learning of a VC-class.
  3. Nearest Neighbor Learning.

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.

References


Ingo Steinwart and Clint Scovel (2005)
Fast Rates for Support Vector Machines
COLT 279-294.


Sanjoy Dasgupta and Daniel Hsu (2008)
Hierarchical sampling for active learning
ICML 2008.


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


Ruth Urner and Shai Ben-David and Shai Shalev-Shwartz (2011)
Unlabeled data can Speed up Prediction Time.
ICML 2011.


Nakul Verma, Samory Kpotufe and Sanjoy Dasgupta (2012)
Which Spatial Partition Trees are Adaptive to Intrinsic Dimension?
CoRR abs/1205.2609.

Experiments with synthetic data

Mixture of Gaussian datasets with the following characteristics

- 80% of the points from 4 dense Gaussian, "centered in the corners" of the space
- 20% of the points from 4 sparse Gaussian centered at random points
- 8 classes - each Gaussian gets a different label

Vary the variance of the Gaussian $\rightarrow$ 3 different datasets

A .1 dense variance and 1 sparse variance

B .01 dense variance and .1 sparse variance

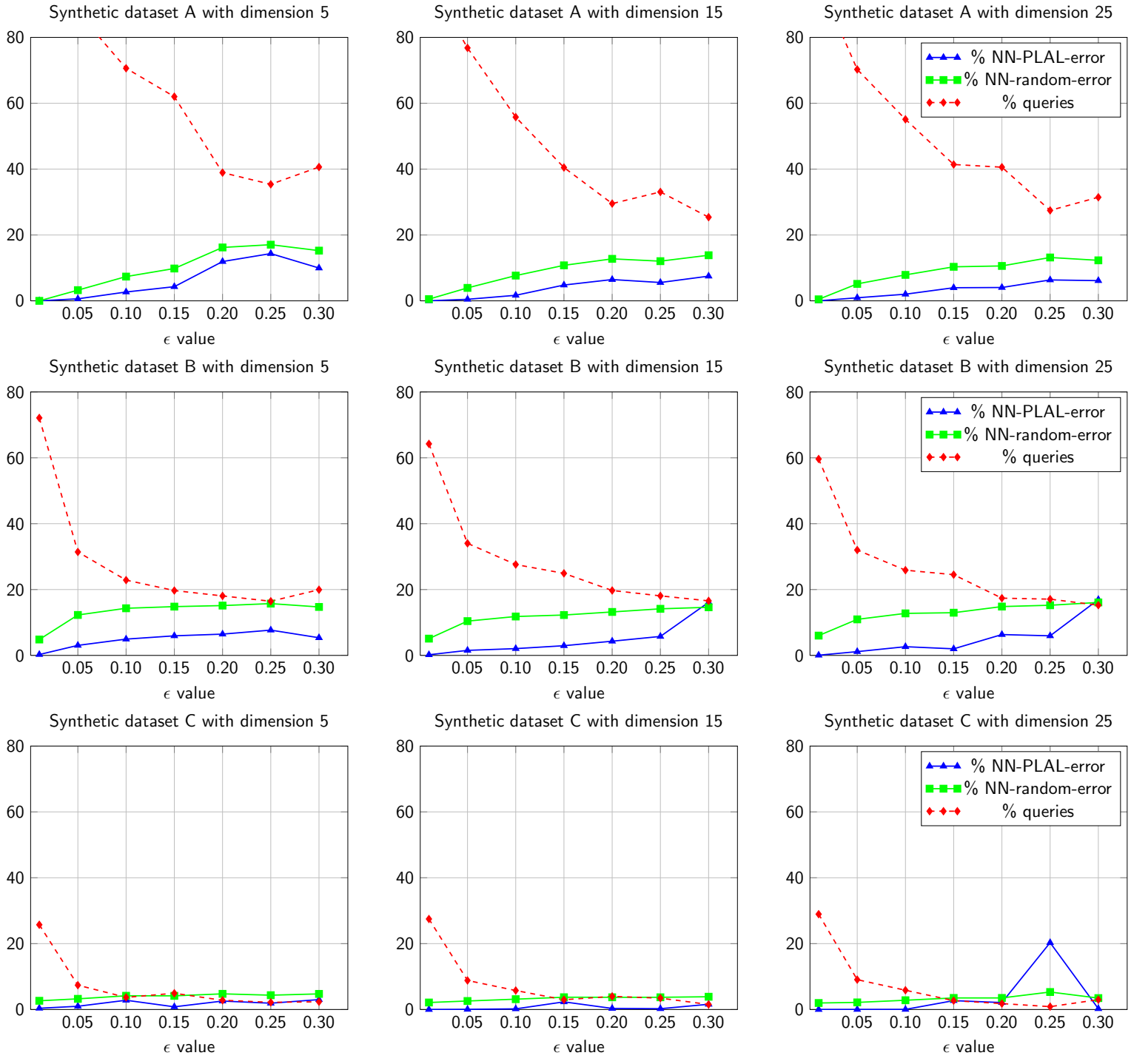
C .001 dense variance and .1 sparse variance

Intuition:
C most cluster able - *A* least cluster able

Prediction error vs. queries

Experiment settings:

We let ϵ range in $(0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3)$. For each ϵ we compute the PLAL queries and predictions, and compare with a k-NN prediction on a random sample of the same size (best k in the range $(1, 3, 5, 10)$).
For each dataset we generate 10 instantiation for each dimension $d = (5, 15, 25)$.


Synthetic dataset A with dimension 5
Synthetic dataset A with dimension 15
Synthetic dataset A with dimension 25
Synthetic dataset B with dimension 5
Synthetic dataset B with dimension 15
Synthetic dataset B with dimension 25
Synthetic dataset C with dimension 5
Synthetic dataset C with dimension 15
Synthetic dataset C with dimension 25
% NN-PLAL-error
% NN-random-error
% queries
ϵ value
0
20
40
60
80
0.05
0.10
0.15
0.20
0.25
0.30

Empirical PL

Settings:

The empirical $\phi(\lambda)$ is calculated as the percentage of data points having at least 1 λ-close neighbor with a different label.