Importance Weighted Active Learning

Alina Beygelzimer
IBM Research
beygel@us.ibm.com

Sanjoy Dasgupta
UC San Diego
dasgupta@cs.ucsd.edu

John Langford
Yahoo Research
jl@yahoo-inc.com

Abstract

We propose an importance weighting framework for actively labeling samples. This technique yields practical yet sound active learning algorithms for general loss functions. Experiments on passively labeled data show that this approach effectively reduces the label complexity required to achieve good prediction performance on many learning problems.

1 Introduction

Active learning is typically defined in contrast to the standard passive learning setting. In passive learning, all of the labels for an unlabeled dataset are requested at once, while in active learning the learning algorithm interactively chooses which unlabeled examples to label. The great hope of active learning is that interaction can substantially reduce the number of labels required, making solving problems via machine learning more practical. This hope is known to be valid in certain special cases where the number of labels required is logarithmic in the number required for passive learning. Canonical special cases include thresholds on a line or linear separators with a spherically uniform unlabeled data distribution [10].

For a long time, active learning algorithms (such as [7, 10]) were not robust to noise, and could even yield inconsistent results (i.e., they are not guaranteed to converge to the optimal predictor, even when given an infinite labeling budget).

This problem has recently been addressed in two threads of research. One approach [3, 9, 12] constructs sample complexity bounds based algorithms satisfying standard PAC-type guarantees for supervised learning. The second uses importance weights to avoid bias due to to active learning [2, 16].

Problems with Existing Active Learning. The PAC-guarantee active learning algorithms have yet to see practical use for a few reasons:

1. In many applications, some loss other than 0–1 loss is important. Yet, these algorithms use the flatness of 0–1 loss with respect to parameters of the hypothesis space to avoid labeling some examples. This technique does not apply to most other loss functions.
2. These algorithms rely on generalization bounds that are often quite loose in practice. When faced with writing a practical active learning algorithm, using these bounds can impose much greater labeling requirements than might be fundamentally necessary.
3. The PAC-guarantee algorithms reason use bounding logic on an entire set of hypotheses which is often computationally intractable (see [8] for an exception for tree-structured hypotheses).

The importance weighted approaches *are* computationally tractable algorithms. However, their analysis and associated guarantees have some drawbacks:

1. The settings in which an importance weighting approach works is limited compared to what we assume here. For example[2] considers linear representations and places some assumptions on the data generation process.
2. The analysis in these works is asymptotic rather than a finite label complexity. Label complexity is of paramount importance in active learning, because otherwise simpler passive learning approaches work fine. Furthermore, choosing the importance weights in a poor manner can result in very poor label complexity performance.

Importance Weighted Active Learning. We address the problems above with a new algorithm satisfying PAC-style label complexity guarantees. The essential idea is to carefully choose rejection sample probabilities for samples, and use derived importance weights to bias the learning process.

To deal with other loss functions, we use the variation in loss amongst remaining hypotheses to create a distribution from which to rejection sample. If the probability of asking for a label is p according to this distribution, then the corresponding importance weight is proportional to $1/p$. This simple method maintains the consistency property for active learning: for any distribution and any hypothesis class, active learning eventually converges to the optimal hypothesis in the class.

The fundamental contribution of this paper is a family of practical general purpose active learning algorithms with bounded label complexity. After introducing basic definitions in section 2, we present an importance weighting algorithm skeleton (IWAL) in section 3, and prove that all algorithms defined by IWAL have a bounded worst-case label complexity to find an ϵ-optimal predictor. Section 4 explores a choice of this subroutine and proves that the resulting algorithm, IWAL(*loss-weighting*), has worst-case label complexity requirements within a constant factor of passive learning. In section 5, we analyze the label complexity, proving both a more general lower bound than earlier work [13] and an upper bound for IWAL(*loss-weighting*) based on a form of the disagreement coefficient [12] generalized to other losses.

We conduct practical experiments with two IWAL algorithms. The first is a specialization of IWAL(*loss-weighting*) to the convex loss with linear representation case where the algorithm becomes tractable via convex programming (section 7). The second IWAL(*bootstrap*) uses a simple bootstrapping scheme, which reduces active learning to (batch) passive learning in a manner which requires only a small amount of additional computation (section 8). These experiments are extremely encouraging: in every case they yield substantial reductions in label complexity compared to passive learning, without compromising predictive performance. These experiments suggest that IWAL is a practically useful realization of the theoretical claim that active learning can reduce the label complexity without compromising prediction performance or safety compared to passive learning[4].

1.1 Additional Prior Work

Naoki Abe and Hiroshi Mamitsuka[1] proposed active learning algorithms based on boosting and bagging which are similar (at the surface) to the IWAL(Bootstrap) algorithm used experimentally in section 8. One critical difference is that these earlier algorithms are not consistent in the presence of adversarial noise: they may never converge to the correct solution, even given an infinite label budget. In contrast, IWAL(Bootstrap) is consistent and satisfies further guarantees (see section 2).

Wiens [17] investigated linear regression in the presence of a contaminating signal added into the target variable. This paper addresses a fundamentally different problem from the one addressed here, since we are actually choosing whether or not to gather and use a target variable, and we are not concerned with a contaminating signal.

2 Preliminaries

Let X be the input space and Y the output space. We consider active learning in the streaming setting where at each step t, a learner observes an unlabeled example $x_t \in X$ and has to decide whether or not to query for the label $y_t \in Y$. The learner works with a hypothesis space $H = \{h : X \to Z\}$, where Z is some prediction space.

The algorithm is evaluated with respect to a given loss function $l : Z \times Y \to [0, \infty)$. The most common 0–1 loss is given by $l(z, y) = \mathbf{1}(y \neq z)$, where $Y = Z = \{-1, 1\}$. The following examples address the binary case $Y = \{-1, 1\}$ with $Z = \mathbb{R}$: $l(z, y) = (1 - yz)_+$ (hinge loss), $l(z, y) = \ln(1 + e^{-yz})$ (logistic loss), $l(z, y) = (y - z)^2 = (1 - yz)^2$ (squared loss), $l(z, y) = |y - z| = |1 - yz|$ (absolute loss). Since any bounded loss function can be normalized, we assume that l maps to $[0, 1]$.

3 The Importance Weighting Skeleton

Algorithm 1 describes the basic outline of importance-weighted active learning (IWAL). IWAL is parameterized by a subroutine *rejection-threshold*, which returns the probability p_t of requesting y_t, given x_t and all previous history $\{x_i, y_i, p_i, Q_i : 1 \leq i < t\}$. Later sections explore different choices for this subroutine. The algorithm maintains a set of labeled examples seen so far, where each example is assigned an importance value. If y_t is queried, IWAL adds $(x_t, y_t, 1/p(t))$ to the set, where $1/p(t)$ is the importance of predicting y_t on x_t.

Algorithm 1 IWAL (subroutine *rejection-threshold*, minimum p_{min})

```
Set $S_0 = \emptyset$.
For $t$ from $1, 2, \ldots$ until the data stream runs out
    1. Receive $x_t$ .
    2. Set $p_t =$ rejection-threshold$(x_t,$ history $\{x_i, y_i, p_i, Q_i : 1 \leq i < t\})$.
    3. Flip a coin $Q_t \in \{0, 1\}$ with $\mathbf{E}[Q_t] = p_t$. If $Q_t = 1$, request $y_t$ and set
           $S_t = S_{t-1} \cup \{(x_t, y_t, p_{\text{min}}/p_t)\}$,
       else $S_t = S_{t-1}$.
    4. Let $h_t = \arg\min_{h \in H} \sum_{(x,y,c) \in S_t} c \cdot l(h(x), y)$.
```

Let D be the underlying probability distribution on $X \times Y$. The expected loss of $h \in H$ on D is given by $L_D(h) = \mathbf{E}_{(x,y) \sim D} l(h(x), y)$. The importance-weighted estimate at time T is

$$L_T(h) = \frac{1}{T} \sum_{t=1}^{T} \frac{Q_t}{p_t} l(f(x_t), y_t).$$

It is not hard to see that $\mathbf{E} L_T(h) = L(h)$, where the expectation is taken over all the random variables involved. However, there is a danger of the variance being high. Theorem 4.1 gives a fairly strong large deviation bound for $L_T(h)$, provided that the probabilities p_t are chosen carefully.

3.1 IWAL Safety

A desirable property of any learning algorithm is *consistency*: Given an infinite budget of unlabeled and labeled examples, does the algorithm converge to the best possible predictor? Several older active learning algorithms [7, 10] do not satisfy this baseline guarantee; in particular, any algorithm requiring realizability is inconsistent in the presence of noise. We prove that IWAL algorithms are consistent, as long as p_t is bounded away from 0. Furthermore, we prove that the label complexity required is within a constant factor of supervised learning in the worst case.

Theorem 3.1. *For all distributions D, for all finite hypothesis classes H, if there is a constant $p_{\text{min}} > 0$ such that $p_t \geq p_{\text{min}}$ for all $1 \leq t \leq T$, then for any $\epsilon > 0$*

$$\mathbf{P}\left[L_T(h) - L(h) > \frac{\sqrt{2}}{p_{\text{min}}} \sqrt{\frac{\ln |H| + \ln \frac{1}{\delta}}{T}} \right] < \delta.$$

where $h^* = \arg\min_{h \in H} L_D(h)$.

Examining this result and comparing with known sample complexity bounds in supervised learning (For example, see [14] Corollary 4.2), we see that the label complexity is at most $\frac{2}{p_{\text{min}}^2}$ times a supervised algorithm's label complexity.

The proof is a simple modification of standard results to use a Martingale inequality. In realistic applications using 32 bit floats to represent real numbers, the hypothesis space is discrete and the bound above is often only a small constant factor worse than a VC-dimension bound. this can also be directly extended to VC spaces by appropriate insertion of a martingale inequality.

Proof. Fix D and omit D from the subscripts. For a hypothesis $h \in H$, consider a sequence of random variables $U_1, \dots, U_T$ with U_t given by

$$U_t = p_{\text{min}} \left(\frac{Q_t}{p_t} l(h(x_t), y_t) - L(h) \right).$$

Since $p_{\text{min}}/p_t \leq 1$, $|U_t| \leq 1$. The sequence $Z_t = \sum_{i=1}^{t} U_i$ is a martingale, letting $Z_0 = 0$. Indeed, for any $1 \leq t \leq T$,

$$\begin{aligned}
\mathbf{E}[Z_t \mid Z_{t-1}, \dots, Z_0] &= \mathbf{E}_{Q_t, x_t, y_t} [U_t + Z_{t-1} \mid Z_{t-1}, \dots, Z_0] \\
&= Z_{t-1} + p_{\text{min}} \cdot \mathbf{E}_{Q_t, x_t, y_t} \left[\frac{Q_t}{p_t} l(h(x_t), y_t) - L(h) \mid Z_{t-1}, \dots, Z_0 \right] \\
&= Z_{t-1} + p_{\text{min}} \cdot \mathbf{E}_{x_t, y_t} [\, l(h(x_t), y_t) - L(h) \mid Z_{t-1}, \dots, Z_0 \,] = Z_{t-1}.
\end{aligned}$$

Observe that $|Z_{t+1} - Z_t| = |U_{t+1}| \leq 1$ for all $0 \leq t < T$. Applying Azuma's inequality,

$$\mathbf{P}[Z_T > \lambda \sqrt{T}] < e^{-\lambda^2/2}$$

for any $\lambda > 0$. Rewriting, we get

$$\mathbf{P}\left[L_T(h) - L(h) > \frac{\lambda}{p_{\text{min}} \sqrt{T}} \right] < e^{-\lambda^2/2}.$$

We want $\lambda / p_{\text{min}} \sqrt{T} < \epsilon/2$, or $\lambda < p_{\text{min}} \epsilon \sqrt{T}/2$.

Applying the union bound, we have for all $h \in H$ simultaneously

$$\mathbf{P}\left[L_T(h) - L(h) > \epsilon/2 \right] < |H| e^{-T p_{\text{min}}^2 \epsilon^2/8}.$$

Changing variables from ϵ to δ, we get that:

$$\mathbf{P}\left[L_T(h) - L(h) > \frac{\sqrt{2}}{p_{\text{min}}} \sqrt{\frac{\ln |H| + \ln \frac{1}{\delta}}{T}} \right] < \delta.$$

□

4 Importance-Weighted Active Learning

Next we instantiate the rejection threshold subroutine in IWAL, and prove that the resulting algorithm IWAL(*loss-weighting*) has several desirable properties. Step 4 of IWAL is modified to do the minimization over H_t instead of H.

IWAL(*loss-weighting*) depends on a sample complexity bound derived quantity:

$$\Delta_t = \sqrt{\frac{8}{t} \ln \frac{t(t+1)|H_t|^2}{\delta}},$$

where t is the index of the sample observed.

4.1 A generalization bound

A fairly strong large deviation bound can be given for each h_t output by IWAL(*loss-weighting*). Note that this theorem is not a corollary of theorem 3.1 because IWAL(*loss-weighting*) can set the importance weight to 0.

Algorithm 2 *loss-weighting* (x, history $\{x_i, y_i, p_i, Q_i : 1 \le i < t\}$)

```
1. Initialize $H_0 = H$.
2. Update
       $L^*_{t-1} = \min_{h \in H_{t-1}} \frac{1}{t-1} \sum_{i=1}^{t-1} \frac{Q_i}{p_i} l(h(x_i), y_i)$
       $H_t = \{h \in H_{t-1} : \frac{1}{t-1} \sum_{i=1}^{t-1} \frac{Q_i}{p_i} l(h(x_i), y_i) \le L^*_{t-1} + \Delta_{t-1}\}$
3. Return $p_t = \max_{f,g \in H_t, y \in Y} l(f(x), y) - l(g(x), y)$.
```

Theorem 4.1. *For all learning problems D, for all hypothesis classes H, for all $\delta > 0$, with probability at least $1 - \delta$, the hypothesis output by* IWAL(*loss-weighting) at any time T satisfies*

$$L(h_T) - \min_{h \in H} L(h) \le 2\Delta_{T-1}.$$

Roughly speaking, this bound shows that the sample complexity of IWAL(*loss-weighting*) is within a constant factor of supervised learning, since $H_{T-1} \subseteq H$. This safety guarantee also suggests that the algorithm can sometimes do much better, because H_{T-1} might be much smaller than H.

The proof of this theorem rests on the following lemma.

Lemma 1. *For all learning problems D, for all hypothesis classes H, for all $\delta > 0$, with probability at least $1 - \delta$, for all T and for all $f, g \in H_T$,*

$$L_T(f) - L_T(g) \le L(f) - L(g) + \Delta_T.$$

Proof. We'll allow failure probability $\delta / T(T+1)$ at time T. Pick any T.

Pick any $f, g \in H_T$. Then $f, g \in H_1, H_2, \ldots, H_{T-1}$. It follows that for all $t \le T$,

$$p_t \ge l(f(x_t), y_t) - l(g(x_t), y_t).$$

Define $Z_t = \frac{Q_t}{p_t}\big(l(f(x_t), y_t) - l(g(x_t), y_t)\big) - (L(f) - L(g))$. Then

$$|Z_t| \le \frac{1}{p_t} |l(f(x_t), y_t) - l(g(x_t), y_t)| + |L(f) - L(g)| \le 2.$$

Also,

$$\begin{aligned}
&\mathbf{E}\left[Z_t \mid Z_1, \ldots, Z_{t-1}\right] \\
&\quad = \mathbf{E}_{x_t, y_t, p_t}\left[\mathbf{E}_{Q_t}\left[\frac{Q_t}{p_t}(l(f(x_t), y_t) - l(g(x_t), y_t)) - (L(f) - L(g)) \,\middle|\, x_t, y_t, p_t\right] \,\middle|\, Z_1, \ldots, Z_{t-1}\right] \\
&\quad = \mathbf{E}_{x_t, y_t, p_t}\left[l(f(x_t), y_t) - l(g(x_t), y_t) - (L(f) - L(g)) \mid Z_1, \ldots, Z_{t-1}\right] = 0.
\end{aligned}$$

Therefore $Z_1, Z_2, \ldots$ is a martingale difference sequence.

(iii) Applying Azuma's inequality to this sequence, we have

$$\begin{aligned}
&\mathbf{P}\left[L_T(f) - L_T(g) \ge L(f) - L(g) + \Delta_T\right] \\
&\quad = \mathbf{P}\left[\frac{1}{T}\left(\sum_{t=1}^{T}\left(\frac{Q_t}{p_t}(l(f(X_t), Y_t) - l(g(X_t), Y_t)) - (L(f) - L(g))\right)\right) \ge \Delta_T\right] \\
&\quad = \mathbf{P}\left[\sum_{t=1}^{T} Z_t \ge T\Delta_T\right] \\
&\quad \le \exp\left(-\frac{T\Delta_T^2}{8}\right) = \frac{\delta}{T(T+1)|H_T|^2}.
\end{aligned}$$

(iv) Now do a union bound over all $f, g \in H_T$. □

Proof. (of Theorem 4.1) Start by assuming that the $1-\delta$ probability event of Lemma 1 holds.

By induction, we can prove that $h^* = \arg\min_{h \in H} L(h)$ satisfies $h^* \in H_T$ for all T.

The base case is $T = 1$, where it clearly holds. Now suppose it holds at T; we'll show it remains true at $T + 1$. Let h_T be the minimizer of $L_T(\cdot)$ over H_T. By Lemma 1,

$$L_T(h^*) - L_T(h_T) \leq L(h^*) - L(h_T) + \Delta_T \leq \Delta_T.$$

Thus $L_T(h^*) \leq L_T^* + \Delta_T$ and hence $h^* \in H_{T+1}$.

Next we show that for any $f, g \in H_T$ we have $L(f) - L(g) \leq 2\Delta_{T-1}$. By Lemma 1,

$$\begin{aligned} L(f) - L(g) &\leq L_{T-1}(f) - L_{T-1}(g) + \Delta_{T-1} \\ &\leq (L_{T-1}^* + \Delta_{T-1}) - L_{T-1}^* + \Delta_{T-1} \;=\; 2\Delta_{T-1}. \end{aligned}$$

Since the optimal hypothesis remains in H_T and the difference in error rates is bounded, the hypothesis h_T must satisfy $L(h_T) \leq L(h^*) + 2\Delta_{T-1}$. □

5 Label complexity

Suppose we see a stream of T examples, some of whose labels we query. The analysis of the previous section tells us that at the end of this process, the final classifier is comparable (in terms of its loss on the underlying distribution) to the classifier that would have been chosen by a supervised learner that saw all T labels.

So, how many of those T labels does the active learner request? Earlier work [9] studied this question under an active learning scheme designed specifically for 0–1 loss. For learning problems with bounded *disagreement coefficient* [12] (to be defined shortly), the number of queries was found to be

$$O(\eta T + d \log^2 T)$$

where d is the VC dimension of the function class, and η is the best error rate achievable on the underlying distribution by that function class.

The term ηT is inevitable for any active learning scheme, as we demonstrate in the next section. The remaining term has just a polylogarithmic dependence on T, which bodes well for active learning.

The method of this paper is substantially more general, because it allows loss functions other than 0–1 loss. To analyze label complexity, we generalize the notion of disagreement coefficient to arbitrary loss functions. Under similar conditions to the earlier result, we find the number of queries is

$$O\left(\eta T + \sqrt{dT \log^2 T}\right),$$

where η is now the best achievable loss by the function class. The inevitable ηT term is still there, and the second term is not as impressive as before, but still sublinear. With more sophisticated generalization bounds, it may be possible to reduce this term, at least for 0–1 loss.

5.1 A lower bound on label complexity

In recent work [13], it was shown that for any nontrivial hypothesis class H and any $\eta > \epsilon > 0$, there exists a data distribution (over $X \times Y$) such that:

- The optimal error rate achievable by H is η.
- Any active learner that finds $h \in H$ with error rate $\leq \eta + \epsilon$ (with probability $> 1/2$) must make η^2/ϵ^2 queries.

We now strengthen this lower bound to $d\eta^2/\epsilon^2$, where d is the VC dimension of H.

Before getting into the details, let's see how this lower bound relates to the label complexity rates mentioned above. It is well-known that if a supervised learner sees T examples (for any $T > d/\eta$), its final hypothesis has error bounded by $\eta + \sqrt{d\eta/T}$ [5]. This is $\eta + \epsilon$ for $\epsilon = \sqrt{d\eta/T}$. The lower bound now implies that an active learner must make at least $d\eta^2/\epsilon^2 = \eta T$ queries. This explains the ηT leading term in all the label complexity bounds we have discussed.

Theorem 5.1. For any $\eta, \epsilon > 0$ such that $2\epsilon \leq \eta \leq 1/4$, for any input space X and hypothesis class H (of functions mapping X into $Y = \{+1, -1\}$) of VC dimension $1 < d < \infty$, there is a distribution over $X \times Y$ such that (a) the best error rate achievable by H is η; (b) any active learner seeking a classifier of error at most $\eta + \epsilon$ must make $\Omega(d\eta^2/\epsilon^2)$ queries to succeed with probability at least $3/4$.

Proof. Pick a set of d points $x_1, x_2, \ldots, x_d$ shattered by H. Here is the distribution over $X \times Y$:

- Point x_1 has probability $1 - \beta$, while each of the remaining x_i has probability $\beta/(d-1)$, where $\beta = 2(\eta + 2\epsilon)$.
- At x_1, the response is always $y = 1$. At $x_i, i > 1$, the response is $y = 1$ with probability $1/2 + \gamma b_i$, where b_i is either $+1$ or -1, and $\gamma = 2\epsilon/\beta = \epsilon/(\eta + 2\epsilon) < 1/4$.

Nature starts by picking $b_2, \ldots, b_d$ uniformly at random. This defines the target hypothesis h^*, where $h^*(x_1) = 1$ and $h^*(x_i) = b_i$. The error rate of h^* is $\beta \cdot (1/2 - \gamma) = \eta$.

An active learner must determine the hidden bit b_i of at least a quarter of the points $x_2, \ldots, x_d$; otherwise (with probability $> 1/2$) it returns a hypothesis that is wrong on at least a quarter of the points x_i and thus has error at least $\eta + (1/4) \cdot \beta \cdot \gamma = \eta + \epsilon$.

To correctly determine a hidden bit b_i with probability $> 1/2$, the learner needs to make $\Omega(1/\gamma^2)$ queries to that x_i. Thus the active learner needs $\Omega(d \cdot (1/\gamma)^2) = \Omega(d\eta^2/\epsilon^2)$ queries in all. □

This is the very same example that is used to give lower bounds on supervised sample complexity (see, for instance, section 14.4 of [11]), although in that case the lower bound is $d\eta/\epsilon^2$. The bound for active learning is smaller by a factor of η because the active learner can avoid making repeated queries to the "heavy" point x_1, whose label is immediately obvious.

5.2 An upper bound on label complexity

This subsection is devoted to proving that IWAL(*loss-weighting*) can yield substantial label complexity improvements over passive learning. We first describe a few concepts: a broader class of loss functions than 0–1 loss, a distance metric on hypotheses bounding the loss difference between these hypotheses, and a generalized disagreement coefficient. We then prove that for this broader class, active learning performs better than passive learning when a generalized disagreement coefficient is small.

5.2.1 A subclass of loss functions

We give label complexity upper bounds for a certain class of loss functions that includes 0–1 loss and logistic loss but not squared loss. Specifically, we require that the loss function has bounded *slope asymmetry*, as defined below.

Recall the earlier notation: the set of classifiers is $H = \{h : X \to Z\}$, where Z is a response space, and the loss function is $l : Z \times Y \to [0, \infty)$. In what follows, the label space is $Y = \{-1, +1\}$.

Definition 1. *The* slope asymmetry *of a loss function* $l : Z \times Y \to [0, \infty)$ *is*

$$C_l = \sup_{z,z' \in Z} \left| \frac{\max_{y \in Y} l(z,y) - l(z',y)}{\min_{y \in Y} l(z,y) - l(z',y)} \right|.$$

Intuitively, the slope asymmetry is the maximum ratio (over choices of truth y) of the derivative of the loss as a function of the prediction. This quantity is generalized to nondifferentiable losses via discrete differences.

It is easy to check that the slope asymmetry is 1 for 0–1 loss, and ∞ for hinge loss. For convex loss functions (of the form $l(z,y) = \phi(yz)$ for convex ϕ) the following lemma helps in bounding C_l.

Lemma 2. *Let ϕ be a differentiable convex function defined on some interval $Z = (-B, B) \subset \mathbb{R}$. Suppose that $C_0 \leq |\phi'(z)| \leq C_1$ for all $z \in Z$. Then for any $z, z' \in Z$, and any $y \in \{-1, +1\}$,*

$$C_0|z - z'| \leq |l_\phi(z,y) - l_\phi(z',y)| \leq C_1|z - z'|$$

where $l_\phi(z,y) = \phi(zy)$. In particular, l_ϕ has slope asymmetry at most C_1/C_0.

Proof. By the mean value theorem, there is some $\xi \in Z$ such that

$$l_\phi(z,y) - l_\phi(z',y) \;=\; \phi(yz) - \phi(yz') \;=\; \phi'(\xi)(yz - yz').$$

Thus $|l_\phi(z,y) - l_\phi(z',y)| = |\phi'(\xi)| \cdot |z - z'|$, and the rest follows from the bounds on ϕ'. □

For instance, this gives a bound on the slope asymmetry of the logistic loss function.

Corollary 1. *The logistic loss function $l(z,y) = \ln(1 + e^{-yz})$, defined on label space $Y = \{-1,+1\}$ and response space $[-B,B]$, has slope asymmetry at most $1 + e^B$.*

Proof. The logistic function ϕ satisfies $1/(1+e^B) \le |\phi'| \le 1$. The rest follows from Lemma 2. □

5.2.2 Topologizing the space of classifiers

We introduce a simple distance function on the space of classifiers.

Definition 2. *For any $f, g \in H$ and distribution D define $\rho(f,g) = \mathbf{E}_{x\sim D} \max_y |l(f(x),y) - l(g(x),y)|$. For any $r \ge 0$, let $B(f,r) = \{g \in H : \rho(f,g) \le r\}$.*

Let $L^* = \min_{h\in H} L(h)$, and suppose it is realized at h^*. We now relate convergence in loss, $L(h_t) \to L(h^*)$, to convergence in the newly topologized classifier space, $\rho(h_t, h^*) \to 0$. The ratio between these rates of convergence can be expressed in terms of the slope asymmetry of the loss function.

Lemma 3. *For any distribution D, if the loss function has slope asymmetry C_l, then*

$$L(h) \le L(h^*) + r \quad \textit{implies} \quad h \in B(h^*, C_l(2L^* + r)).$$

Proof. Pick any $h \in H$ with $L(h) \le L(h^*) + r$.

$$\begin{aligned}
\rho(h,h^*) &= \mathbf{E}_{x\sim D} \max\nolimits_y |l(h(x),y) - l(h^*(x),y)| \\
&\le C_l\, \mathbf{E}_{x,y\sim D} |l(h(x),y) - l(h^*(x),y)| \\
&\le C_l\; (\mathbf{E}_{x,y\sim D}[l(h(x),y)] + \mathbf{E}_{x,y\sim D}[l(h^*(x),y)]) \\
&= C_l\,(L(h) + L(h^*)) \;\le\; C_l(2L^* + r).
\end{aligned}$$

□

5.2.3 A generalized disagreement coefficient

When analyzing the A^2 algorithm [3] for active learning under 0–1 loss, Hanneke found [12] that its label complexity could be characterized in terms of what he called the *disagreement coefficient* of the learning problem. We now generalize this notion to arbitrary loss functions.

Definition 3. *The disagreement coefficient is the infimum value of θ such that for all r,*

$$\mathbf{E}_{x\sim D} \sup\nolimits_{h\in B(h^*,r)} \sup\nolimits_y |l(h(x),y) - l(h^*(x),y)| \;\le\; \theta r.$$

Here is a simple example for linear separators under convex loss.

Lemma 4. *Suppose H consists of linear classifiers $\{u \in \mathbb{R}^d : \|u\| \le B\}$ and the data distribution D is uniform over the surface of the unit sphere in $\mathbb{R}^d$. Suppose the loss function is $l(z,y) = \phi(yz)$ for convex ϕ with $C_0 \le |\phi'| \le C_1$. Then the disagreement coefficient is at most $(C_1/C_0)\sqrt{d}$.*

Proof. Let h^* be the optimal classifier, and h any other classifier with $\rho(h,h^*) \le r$. Let u^*, u be the corresponding vectors in $\mathbb{R}^d$. Using Lemma 2,

$$\begin{aligned}
r \;&\ge\; \mathbf{E}_{x\sim D} \sup_y |l(h(x),y) - l(h^*(x),y)| \;\ge\; C_0\, \mathbf{E}_{x\sim D} |h(x) - h^*(x)| \\
&= C_0\, \mathbf{E}_{x\sim D} |(u - u^*) \cdot x| \;\ge\; C_0\, \frac{\|u - u^*\|}{\sqrt{d}}.
\end{aligned}$$

Thus for any $h \in B(h^*, r)$, we have that the corresponding vectors satisfy $\|u - u^*\| \leq r\sqrt{d}/C_0$. We can now bound the disagreement coefficient.

$$\begin{aligned}
&\mathbf{E}_{x\sim D} \sup_{h\in B(h^*,r)} \sup_y |l(h(x), y) - l(h^*(x), y)| \\
&\quad\leq\quad C_1\,\mathbf{E}_{x\sim D} \sup_{h\in B(h^*,r)} |h(x) - h^*(x)| \\
&\quad\leq\quad C_1\,\mathbf{E}_{x\sim D} \sup\{|(u-u^*)\cdot x| : \|u-u^*\| \leq r\sqrt{d}/C_0\} \;\leq\; C_1 \cdot \frac{r\sqrt{d}}{C_0}.
\end{aligned}$$

□

5.2.4 Upper bound on label complexity

Finally, we give an upper bound on label complexity for learning problems that have bounded disagreement coefficient, and loss functions that have bounded slope asymmetry. Recall that the algorithm sees a stream of T unlabeled examples and chooses to query some of their labels.

Theorem 5.2. *For all learning problems D and hypothesis spaces H, if the loss function has slope asymmetry C_l, and the learning problem has disagreement coefficient θ, then the expected number of labels requested by IWAL(loss-weighting) during the first T iterations is at most*

$$4\theta \cdot C_l \cdot \left(L^*T + O\left(\sqrt{T \ln \frac{|H|T}{\delta}}\right)\right),$$

where L^* is the minimum loss achievable on D by H.

Proof. Let h^* be an optimal classifier in H, achieving loss L^*. Pick any time t. By Lemma 1, $H_t \subset \{h \in H : L(h) \leq L^* + 2\Delta_{t-1}\}$. Thus, by Lemma 3, $H_t \subset B(h^*, r)$ for $r = C_l(2L^* + 2\Delta_{t-1})$.

The expected value of p_t (over the choice of x at time t) is at most

$$\begin{aligned}
\mathbf{E}_{x\sim D} \sup_{f,g\in H_t} \sup_y |l(f(x), y) - l(g(x), y)| \quad&\leq\quad 2\,\mathbf{E}_{x\sim D} \sup_{h\in H_t} \sup_y |l(h(x), y) - l(h^*(x), y)| \\
&\leq\quad 2\,\mathbf{E}_{x\sim D} \sup_{h\in B(h^*,r)} \sup_y |l(h(x), y) - l(h^*(x), y)| \\
&\leq\quad 2\theta r \;=\; 4\theta \cdot C_l \cdot (L^* + \Delta_{t-1}).
\end{aligned}$$

Summing over $t = 1, \ldots, T$, we get the lemma. □

Section 2 in the appendix gives an example showing that it is possible to achieve substantial label complexity reductions over passive learning even when the slope asymmetry is infinite.

6 Other Examples of Low Label Complexity

It's also sometimes possible to achieve substantial label complexity reductions over passive learning, even when the slope asymmetry is infinite.

Example 1. *Let the space X be the ball of radius 1 in d dimensions.*

Let the distribution D on X be a point mass at the origin with weight $1-\beta$ and label 1 and a point mass at $(1, 0, 0, ..., 0)$ with weight β and label -1 half the time and label 0 for the other half the time.

Let the hypothesis space be linear with weight vectors satisfying $||w|| \leq 1$.

Let the loss of interest be squared loss: $l(h(x), y) = (h(x)-y)^2$ which has infinite slope asymmetry.

Observation 6.1. *For the example above, IWAL(loss-weighting) requires only an expected β fraction of the labeled samples of passive learning to achieve the same loss.*

Proof. Passive learning samples from the point mass at the origin a $(1-\beta)$ fraction of the time, while active learning only samples from the point mass at $(1,0,0,...,0)$ since all predictors have the same loss on samples at the origin.

Since all hypothesis h have the same loss for samples at the origin, only samples not at the origin influence the sample complexity. Active learning samples from points not at the origin $1/\beta$ more often than passive learning, implying the theorem. □

7 Implementation

We now explain how to efficiently implement the learning algorithm in the important case where H is the class of bounded-length linear separators $\{u \in \mathbb{R}^d : \|u\|^2 \leq B\}$ and the loss function is convex: $l(z,y) = \phi(yz)$ for convex ϕ.

Each iteration of Algorithm 2 involves solving two optimization problems, both of which are defined over a restricted hypothesis set

$$H_t \;=\; \bigcap_{t'<t} \left\{ h \in H : \frac{1}{t'} \sum_{i=1}^{t'} \frac{Q_i}{p_i} l(h(x_i), y_i) \leq L^*_{t'} + \Delta_{t'} \right\}.$$

Replacing each hypothesis h by its corresponding vector u, this becomes

$$H_t \;=\; \bigcap_{t'<t} \left\{ u \in \mathbb{R}^d : \|u\|^2 \leq B \text{ and } \frac{1}{t'} \sum_{i=1}^{t'} \frac{Q_i}{p_i} \phi(u \cdot (y_i x_i)) \leq L^*_{t'} + \Delta_{t'} \right\}.$$

Since ϕ is convex, this feasible region H_t is an intersection of convex constraints.

The first optimization problem in Algorithm 2 computes $L^*_T = \min_{u \in H_T} \sum_{i=1}^T \frac{Q_i}{p_i} \phi(u \cdot (y_i x_i))$. This is a convex program.

The second optimization problem is $\max_{u,v \in H_T} \phi(y(u \cdot x)) - \phi(y(v \cdot x)),\ y \in \{+1,-1\}$ (where u, v correspond to functions f, g). It is not hard to see that if ϕ is nonincreasing in its argument (as it is for 0–1 loss, or hinge loss, or logistic loss), then the solution of this problem is $\max\{\phi(-A(-x)) - \phi(A(x)), \phi(-A(x)) - \phi(A(-x))\}$, where $A(x)$ is the solution a convex program: $A(x) \equiv \min_{u \in H_T} u \cdot x$. The two cases inside the max correspond to the choices $y = 1$ and $y = -1$, respectively.

Thus Algorithm 2 can be efficiently implemented for nonincreasing convex loss functions and bounded-length linear separators.

In our experiments below, we use an even simpler implementation. For the first problem (determining L^*_T), we minimize over H rather than H_T. The final hypothesis h_T is then consistent (by Theorem 3.1) but doesn't necessarily satisfy the large deviation bound of Theorem 4.1 unless this h_T happens to lie in H_T. For the second problem, instead of using $\min_{u \in H_T} u \cdot x$, in which H_T is defined by $T-1$ convex constraints, we simply enforce the very last of these constraints (which corresponds to time $T-1$). This yields a larger feasible region for u and could thus lead to an overly conservative choice of p_t; but once again, the consistency of h_T is assured.

7.1 Experiments

Recent consistent active learning algorithms [3, 9] have suffered from problems of computational intractability. This section shows that importance weighted active learningis practical.

We implemented IWAL(*loss-weighting*) for linear separators under logistic loss. As outlined above, the algorithm involves two convex optimizations as subroutines. These were coded using log-barrier methods (section 11.2 of [6]). We tried out the algorithm on the MNIST data set of handwritten digits by picking out the 3's and 5's as two classes, and choosing 1000 exemplars of each for training and another 1000 of each for testing. We used PCA to reduce the dimension from 784 to 25. The algorithm uses a generalization bound Δ_t of the form $\sqrt{d/t}$; since this is believed to often be loose in high dimensions, we also tried a more optimistic bound of $1/\sqrt{t}$. In either case, active learning achieved very similar performance (in terms of test error or test logistic loss) to a supervised learner that saw all the labels. The active learner asked for less than a third of the labels. More details are in the appendix.

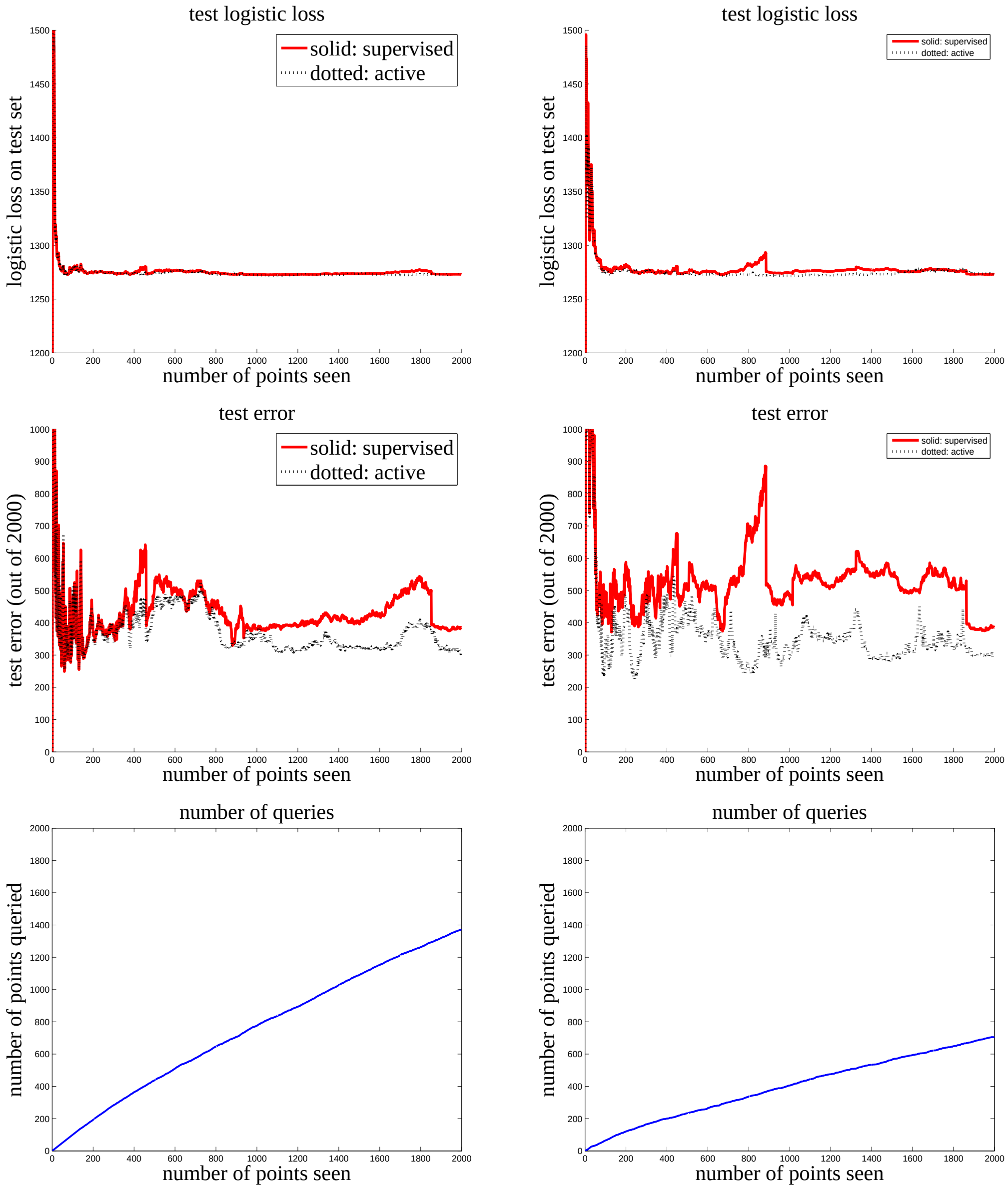

Figure 1: The right column corresponds to the same setup and data set as the left column, but using a more optimistic generalization bound (described in section 6.1). Top: Logistic loss on test set, as the number of points seen grows from 0 to 2000. The solid line is supervised learning and the dotted line is active learning. Middle: A similar plot for the test error. In both cases, supervised and active learning give very similar results. Bottom: Number of queries is sublinear in the number of points seen.

8 Bootstrap instantiation of IWAL

This section discusses another practical implementation of the IWAL skeleton, which fills in the rejection threshold subroutine with a very simple bootstrapping scheme.

Algorithm 3 *bootstrap* (unlabeled example x_t, history)

```
Parameters: $b$ (the length of the initial bootstrap sample), $k$ (the number of predictors used in voting), $p_{\min}$ (a lower bound on the rejection threshold)
    • If $t \leq b$, set $p_t = 1$.
    • If $t = b$, train predictors $h_1, ..., h_k$ on the initial sample $(x_1, y_1, c_t), \ldots, (x_b, y_b, c_t)$. Denote the set by $H$.
    • If $t > b$, set $p_t = p_{\min} + (1 - p_{\min}) \left[ \max_{y, h_i \in H, h_j \in H} L(h_i(x), y) - L(h_j(x), y) \right]$.
    • Return $p_t$.
```

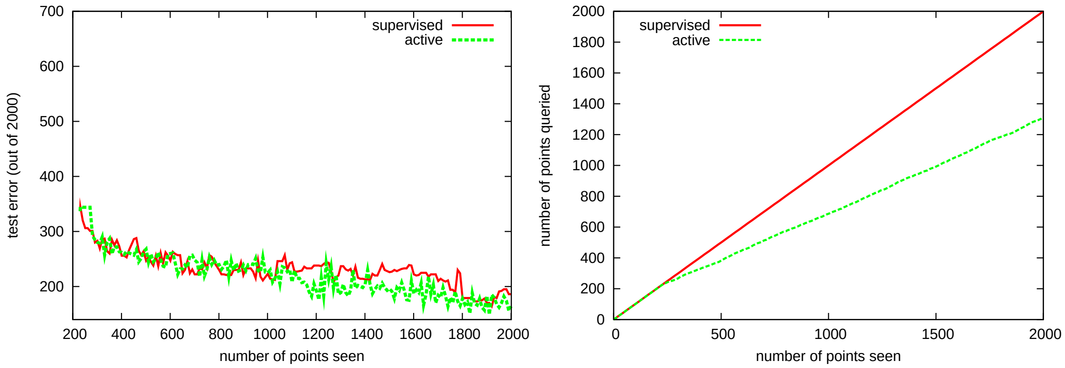

Figure 2: Experiments with *bootstrap*. Left: Test error, as the number of unlabeled points seen grows from 200 (the size of the initial batch, where active learning queries every label) to 2000. Right: Number of queries as a function of the number of points seen.

The *bootstrap* routine in Algorithm 3 simplifies Algorithm 2 by replacing the version space with an approximate version space given by bootstrap trained predictors. Since the version space is only approximate, the rejection threshold is lower bounded by $(1-p_{\min})$. The optimization on the collected importance-weighted training set can be done using any importance-weighted passive learning algorithm for the desired loss function. Thus, *bootstrap* reduces active learning to importance-weighted *batch* passive learning. The latter can be further reduced to batch passive learning using some general conversion mechanism such as Costing [18].

The results of experiments with the bootstrapping scheme are reported in Figure 2. The same MNIST dataset was used as in section 6.1. We implemented *bootstrap* for 0–1 loss with the following parameters: the initial sample contained 1/10th of the training dataset (b=200), $k = 10$, and $p_{\min} = 0.1$. The Costing technique [18] was used to reduce from importance-weighted binary classification to binary classification. (The same technique can be applied to any loss function.) The induced binary classification problem was then solved using a decision tree learner (J48) from Weka [19].

For simplicity, we did't retrain the predictors for each queried point, i.e., the predictors are trained once on an initial batch of b examples. The final predictor is trained on the collected importance-weighted training set. IWAL(*bootstrap*) performed similarly to passive learning with J48, using only two-thirds of the labels.

Unless stated otherwise, the following experiments use the Costing reduction to remove the importances, and trees.J48 (in weka) as a base classifier learner. The set H (with 10 classifiers) is trained only once after bootstrapping on 10% of the training set. Each experiment was run once; all runs tested are reported.

References

[1] Naoki Abe and Hiroshi Mamitsuka. Query Learning Strategies Using Boosting and Bagging. ICML 1998, page 1-9.

[2] Francis Bach. Active Learning for Misspecified Generalized Linear Models. NIPS19.

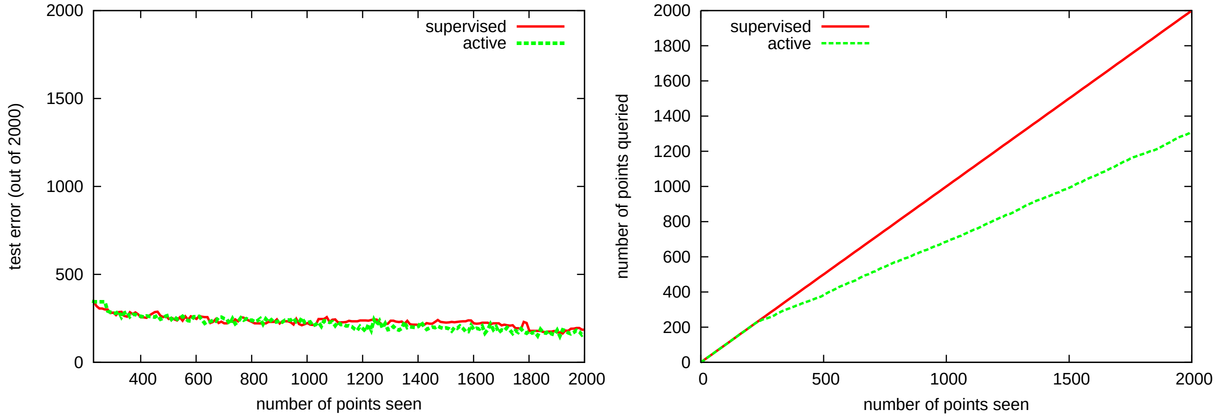

Figure 3: A subset of the **mnist** dataset (binary, 3 versus 5): 2000 training and 2000 test examples, bootstrapped on the initial 10%. Queried 65.6%.

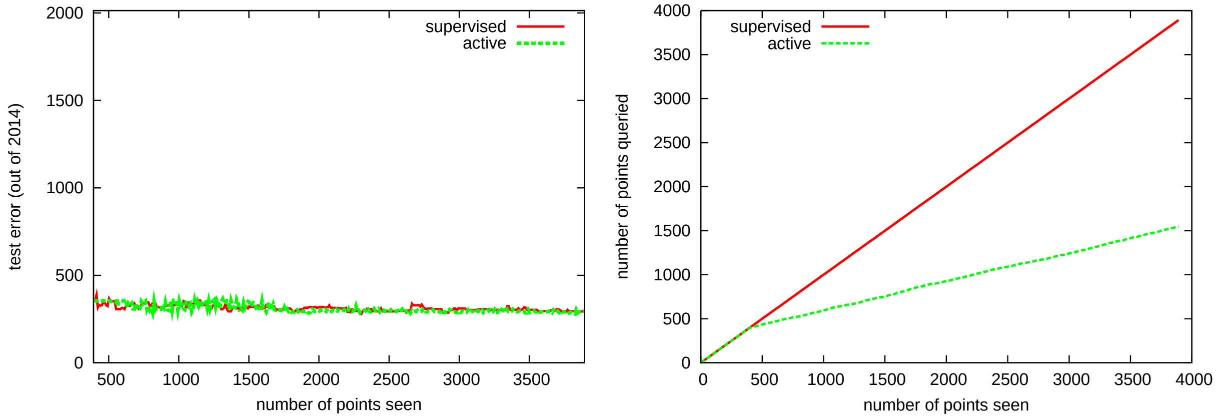

Figure 4: **adult** dataset (binary): roughly 4000 training and 2000 test examples, bootstrapped on the initial 10%. Queried 40.0%.

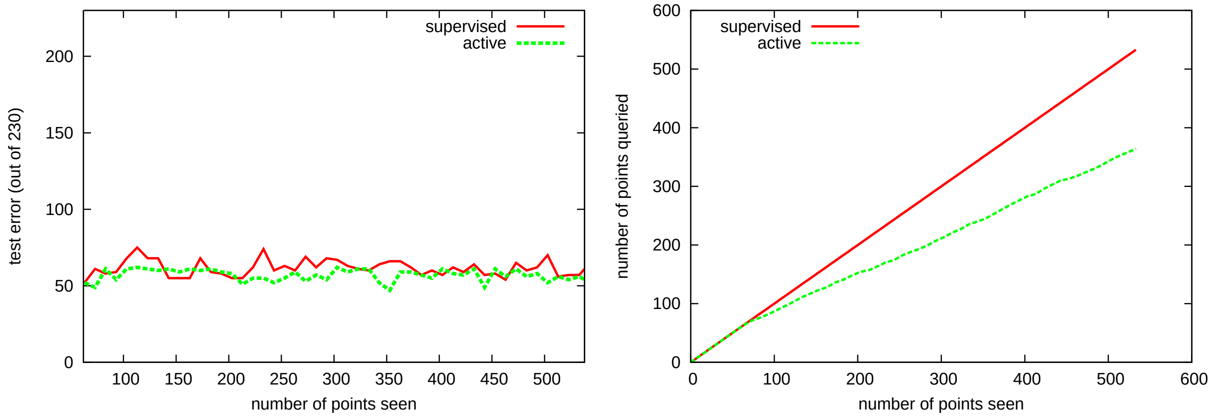

Figure 5: **pima** dataset (binary): 538 training and 230 test examples, bootstrapped on the initial 10%. Queried 67.6%.

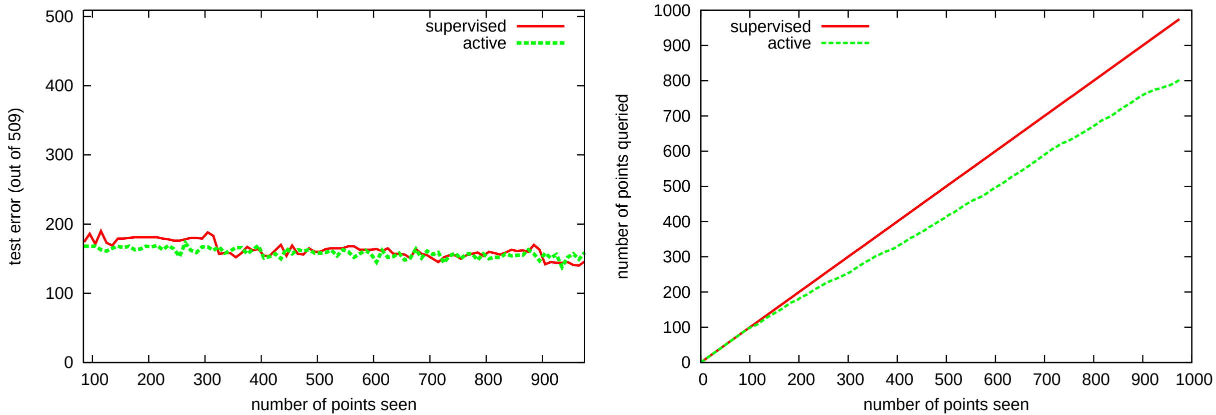

Figure 4: **yeast** dataset (binary): roughly 1000 training and 500 test examples, bootstrapped on the initial 10%. Queried 82.2%.

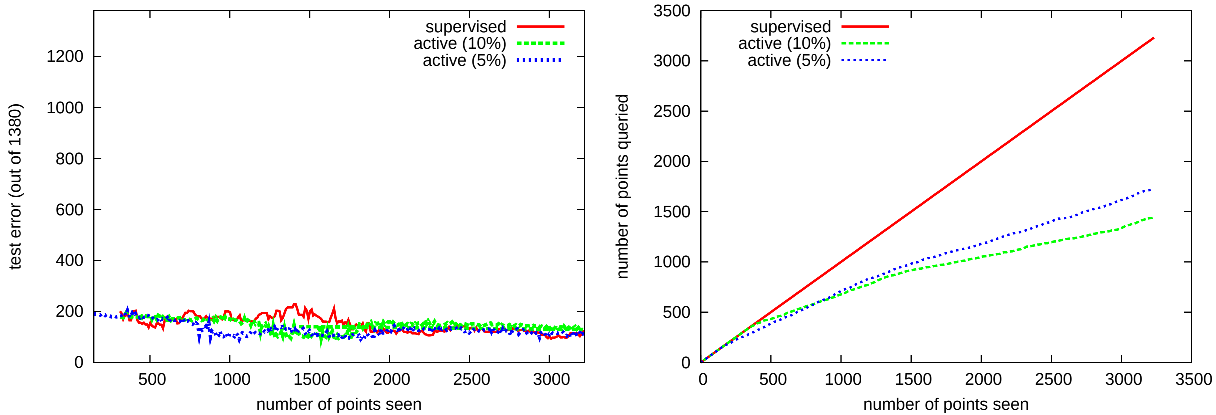

Figure 5: **spambase** dataset (binary): 3221 training and 1380 test examples, bootstrapped on the initial 10% and 5%. Queried 44.2% and 53.5% respectively.

[3] Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. *In proceedings of the 23rd international conference on Machine learning* (ICML), 2006.

[4] Balcan, M.-F., Hanneke, S., Wortman, J. The True Sample Complexity of Active Learning. 21st Annual Conference on Learning Theory (COLT), 2008.

[5] Olivier Bousquet, Stephane Boucheron, and Gabor Lugosi. Introduction to statistical learning theory. *Lecture Notes in Artificial Intelligence, 3176*, 169–207, 2004.

[6] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[7] David Cohn, Les Atlas, and Richard Ladner. Improving Generalization With Active Learning. Machine Learning, 15(2):201-221, 1994.

[8] S. Dasgupta and D.J. Hsu. Hierarchical sampling for active learning. Twenty-Fifth International Conference on Machine Learning (ICML), 2008.

[9] Sanjoy Dasgupta, Daniel Hsu, and Claire Monteleoni, A general agnostic active learning algorithm. Neural Information Processing Systems (NIPS), 2007.

[10] Sanjoy Dasgupta, Adam Kalai, and Claire Monteleoni. Analysis of Perceptron-Based Active Learning. COLT, 2005.

[11] Luc Devroye, Laszlo Gyorfi, and Gabor Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, 1996.

[12] Steve Hanneke. A Bound on the Label Complexity of Agnostic Active Learning. *In proceedings of the 24th Annual International Conference on Machine Learning* (ICML), 2007.

[13] Matti Kaairiainen. On Active Learning in the Nonrealizable Case. ALT, 2006.

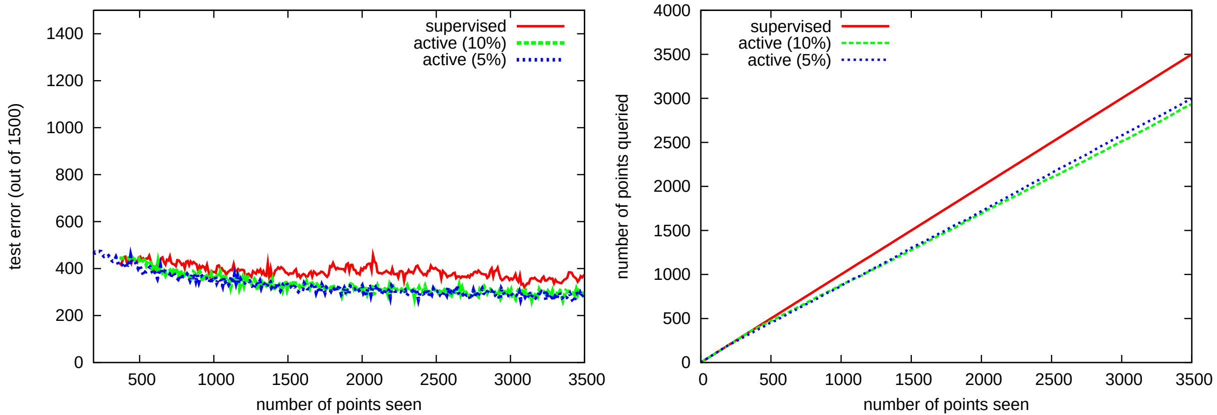

Figure 6: **waveform** dataset (3 classes): 3500 training and 1500 test examples, bootstrapped on the initial 10% and 5%. Queried 83.7% and 85.6% respectively.

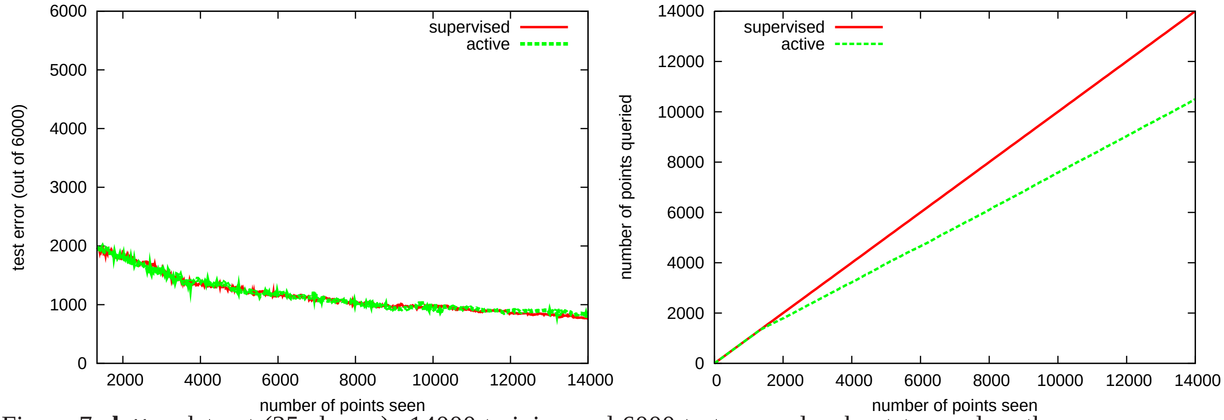

Figure 7: **letter** dataset (25 classes): 14000 training and 6000 test examples, bootstrapped on the initial 10%. Queried 75.0%.

[14] John Langford. Practical Prediction Theory for Classification. JMLR 6(Mar):273–306, 2005.

[15] H. Shimodaira, Improving Predictive Inference under Covariate Shift by Weighting the Log-Likelihood Function, Journal of Statistical Planning and Inference, 227-244:90(2), 2000.

[16] M. Sugiyama Active Learning for Misspecified Models, NIPS 18.

[17] D.P. Wiens. Robust Weights and Designs for Biased Regression Models: Least Squared and Generalized M-Estimation, Journal of Statistical Planning and Inference, 395-412:83(2), 2000.

[18] Bianca Zadrozny, John Langford, and Naoki Abe. Cost-Sensitive Learning by Cost-Proportionate Example Weighting. In *Proceedings of the 3rd IEEE International Conference on Data Mining* (ICDM) 435–442, 2003.

[19] Weka, `http://www.cs.waikato.ac.nz/ml/weka/`.