Algorithms for Active Learning

Daniel Joseph Hsu

June 2, 2010

Contents

Chapter 1

Introduction

We present an overview of active learning and the contributions of this dissertation.

1.1 Active and Passive Learning

This dissertation is concerned with automated procedures for active learning. Active learning is best described in contrast to passive learning—a standard, well-studied learning framework established in statistics and machine learning. In passive learning (sometimes referred to simply as supervised learning), the goal of a learner is to infer an accurate predictor from labeled training data. The labeled training data are examples of input-output pairs (x, y): the output (or label) y represents the correct answer to a question associated with the input x. For example, in the problem of e-mail classification, the label y may be the "yes" / "no" answer to whether a particular e-mail message x is spam or not. These labeled examples are collected prior to the learning (training) process, and the intention is to deploy a learned predictor to predict the labels of input instances x encountered in the future. The goal of the learner, during the training process, is to infer such a predictor from the training data that is accurate with respect to these future input instances.

Active learning models a slightly different framework in which the initially available data does not come with any labels. That is, each training data point is simply an input x without an associated label y. The goal of the active learner is the same as that of a passive learner: to infer an accurate predictor of labels from inputs. However, the active learner is allowed to request the label y of any particular input x in the training data; these requests can be made sequentially, so as to adapt to the results of previous label requests. In the e-mail classification example, this function of the active learner can be seen as asking the user whether a particular e-mail in the mailbox is spam or not. This interactive process of building up a (partially) labeled data set may continue for some time, but eventually a predictor must be returned by the active learner for use in predicting the labels of future input instances.[1]

The practical motivations of the active learning framework are grounded in the disparity

[1] An intermediate framework between supervised (passive) learning and active learning is called semi-supervised learning [CSZ06]. There, the learner is given both labeled data and unlabeled data (and typically the latter is in relative abundance), but otherwise the learning process is the same as passive learning.

between the availability of labeled and unlabeled data. Unlabeled data is nowadays often available in vast quantities, with the raw features of input instances easily collected by automatic processes. For instance, the internet contains trillions of web pages that are readily collected by robots. However, assigning a label to a web page (say, of the page's subject matter) may demand significantly more effort. Labeling typically requires some manual intervention to evaluate or judge input instances, and this can be a costly enterprise (*e.g.*, in terms of time or money), especially relative to the high-throughput collection of the unlabeled data itself. Therefore, in many modern applications of machine learning, only unlabeled data is available cheaply and in large quantities, whereas the labels are expensive to obtain.

The active learning framework addresses the challenge faced in these modern applications by explicitly modeling the process of obtaining labels for unlabeled data. The hope is that the active learner just needs to request the labels of just a few, carefully chosen points during the interactive process in order to produce an accurate predictor.

This dissertation explores both the algorithmic and statistical aspects of active learning for binary classification. What are effective procedures for determining which data to label? How can these procedures take advantage of the interactive learning process, and in what circumstances do they yield improved learning performance compared to standard passive learners? To answer these questions, we develop and rigorously analyze a broad class of general active learning methods that address the essential algorithmic and statistical difficulties of the problem.

1.2 Some Motivating Examples

Learning Threshold Functions

Consider first the task of learning a threshold function of a single variable. A single-variable threshold function $f_\theta : \mathbb{R} \to \{\pm 1\}$, parameterized by the real number $\theta \in \mathbb{R}$ (the threshold value), is defined by

$$f_\theta(x) := \begin{cases} +1 & \text{if } x \geq \theta \\ -1 & \text{if } x < \theta \end{cases}$$

for all $x \in \mathbb{R}$. Threshold functions are a basic tool for classifying univariate data.

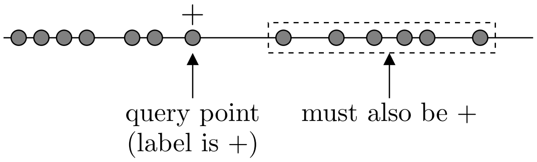
Suppose a (passive) learner is presented with n labeled examples, *i.e.*, pairs $(x_i, y_i) \in \mathbb{R} \times \{\pm 1\}$ for $1 \leq i \leq n$. A reasonable predictor that the learner could produce is one for which the number of disagreements with the given examples is minimal. That is, the learner could choose $\theta \in \mathbb{R}$ such that

$$|\{1 \leq i \leq n : f_\theta(x_i) \neq y_i\}|$$

is as small as possible. For now, we assume that all of the labels actually correspond to some threshold function f_{θ^*}, so $y_i = f_{\theta^*}(x_i)$ for all $1 \leq i \leq n$. Therefore, the learner can easily find some threshold value $\theta \in \mathbb{R}$ that has no disagreements with the given examples, so $|\{1 \leq i \leq n : f_\theta(x_i) \neq y_i\}| = 0$.

Suppose now the same examples are presented to an active learner, except that the labels y_i are initially withheld. It turns out that an active learner can also find a threshold value

$\theta \in \mathbb{R}$ such that f_θ has no disagreements with the (x_i, y_i), and it can do so after requesting just $\log_2 n$ of the labels! To see this, note the correspondence of this problem to binary search for the target threshold θ^*: if a requested label y_i is $+1$, then we can infer that $\theta^* \leq x_i$, and therefore $y_j = +1$ for all $x_j \geq x_i$; if y_i is -1, then $\theta^* > x_i$, and therefore $y_j = -1$ for all $x_j \leq x_i$. Thus, one can simply choose to request the label of a point x_i at the median of the unlabeled points; this is guaranteed to result in an outcome that lets the learner label (for free) at least half of the other unlabeled points.



Active Learning as Binary Search?

The strategy for learning single-variable threshold functions represents a best-case scenario for active learning: just $\log_2 n$ label requests are needed to deduce all of the n labels, after which standard passive learning techniques (such as returning a consistent predictor) can be readily applied. What aspects of the learning problem made this possible?

1. At any point in the interactive process, the active learner could always make a query (label request) that results in labeling (for free) at least half of the other unlabeled points. Viewed another way, the query eliminates at least half of the potential classifiers still in contention.

2. We crucially made an assumption that the labels $y_i = f_{\theta^*}(x_i)$ correspond to some threshold function f_{θ^*}.

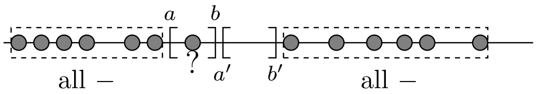
Unfortunately, these aspects do not always carry over to other learning problems: there need not always be queries that provide the information needed for a binary search-like process, even when the labels perfectly correspond to a simple function. And, of course, labels are often noisy, whether due to the occasional erroneous annotation or because of model mismatch.

Learning Interval Functions

Consider now the problem of learning single-variable interval functions $f_{a,b} : \mathbb{R} \to \{\pm 1\}$, where

$$f_{a,b}(x) := \begin{cases} +1 & \text{if } a \leq x < b \\ -1 & \text{if } x < a \text{ or } x \geq b. \end{cases}$$

Even in the case where the labels correspond exactly to some interval function f_{a^*,b^*}, the active learner may need to request all labels in order to distinguish between intervals that include any particular x_i (*i.e.*, one for which $f_{a,b}(x_i) = +1$), and an interval that includes none of the x_i (*i.e.*, one for which $f_{a,b}(x_i) = -1$ for all $1 \leq i \leq n$) [Das05]. In the example depicted below, all of the boxed points are determined to be -1, and still the active learner cannot avoid requesting the label of the final point to choose between $f_{a,b}$ and $f_{a',b'}$.



Thus, the active learning process need not take the form of a straightforward binary search.

Consider the following two-phase strategy for learning a single-variable interval function $f_{a,b}$, also described in [Das05].

1. Request the label of randomly chosen x_i until some y_i is found such that $y_i = +1$. If no $y_i = +1$, then return the empty interval function.

2. Use the binary search-like procedure for learning single-variable threshold functions to determine the interval boundaries a and b, and return $f_{a,b}$.

The crucial observation behind this algorithm is that an interval function can be described by two single-variable threshold functions

$$f_{a,b}(x) = \begin{cases} +1 & \text{if } f_a(x) = +1 \quad \text{and} \quad f_b(x) = -1 \\ -1 & \text{if } f_a(x) = -1 \quad \text{or} \quad f_b(x) = +1. \end{cases}$$

The binary search for b pretends that all points to the left of positive point x_i have a negative label; the binary search for a is similar.

The first phase of the algorithm is certainly not like binary search, but it serves the useful purpose of identifying a starting point for binary search in the second phase. In the worst case, the algorithm may end up querying every label before transitioning into this second phase. But if a significant fraction of the points are labeled +1 by f_{a^*,b^*}, then the first phase ends quickly.

Both phases of the algorithm are susceptible to noise. Can it be made more robust? Suppose it is known that

$$|\{1 \leq i \leq n : f_{a^*,b^*}(x_i) \neq y_i\}| \;\leq\; \frac{1}{2} \cdot |\{1 \leq i \leq n : f_{a^*,b^*}(x_i) = +1\}|.$$

Then, the first phase is modified so that instead of using the first positively-labeled point as the basis for the start of the binary searches for a and b, we use the median of the first several positively labeled points (the precise number depends on the level of confidence desired). Note that this median point will be positively labeled by f_{a^*,b^*} as long as a majority of the positively labeled points are as well. Therefore, this modified procedure produces a point that reduces the task to that of active learning threshold functions. Although this was not a complete nor general solution, it suggests that with some care, active learning methods can in fact be made robust to noise.

General Procedures

Instead of developing specific procedures for each individual learning problem of interest (*e.g.*, a special procedure for learning of threshold functions, a different procedure for learning interval functions, and so on), we will develop general methods that tackle broad classes of learning problems together. Of course, this approach cannot yield better solutions for individual problems than specialized methods. However, by approaching the problem of active learning from a more abstract perspective, we can identify general issues specific to active learning algorithms that are distinguished from the concerns of passive learning. The general procedures that are developed can then be specialized to specific problems with fine tuning that is often anyway required in practice.

1.3 Literature Review

The review in this section focuses primarily on algorithmic techniques for active learning that have been rigorously analyzed. For a review of various heuristic techniques, applications, and model extensions, see the survey of [Set09].

The theoretical study of active learning for binary classification initially focused on a model of learning with membership queries [Ang98, Ang04]. In this model, the learner is allowed to query the label of any unlabeled data point, even an artificially created one. The primary drawback of this model is that the synthesized data points may be too unnatural for a human to label them [LB92]. Therefore, the theoretical focus of active learning has turned to a model in which the learner is only allowed to query the label of data points drawn from the underlying distribution.

The work of Cohn, Atlas, and Ladner [CAL94] (which will be discussed in Chapter 2) presented a selective sampling scheme based on uncertainty sampling in noise-free settings. This scheme has been the inspiration for many subsequent work on active learning, including the algorithms developed in this dissertation. The idea of uncertainty sampling—querying the label of points about which the learner is least sure about—quickly took on many forms, both in probabilistic and non-probabilistic settings [LC94, LG94, SC00]. The query-by-committee (QBC) algorithm of Seung, Opper, and Sompolinsky [SOS92] considered a form of uncertainty sampling grounded in a Bayesian framework, where uncertainty is measured relative to a prior distribution over hypotheses or models. QBC was formally analyzed in [FSST97], where it was shown that the class of linear separators under a uniform data distribution could be learned exponentially faster in the active learning model than in a passive learning model. A simpler (non-Bayesian) algorithm for this task was given in [DKM05].

In the noise-free setting, active learning was abstractly studied by Dasgupta in [Das04, Das05]. The work in [Das04] analyzed a greedy algorithm that is often approximated by Bayesian methods (*e.g.*, [TK00]). It also initiated the rigorous study of the generalization properties of active learning algorithms. The work in [Das05] characterized the label complexity of active learning problems with a parameter called the splitting index—upper and lower bounds on label complexity were proved in terms of this quantity. Unfortunately, the algorithm achieving the upper bound is generally intractable. A different perspective on sample complexity considered in [BHW08] shows that active learning always strictly improves on the label complexity of passive learning, although the improvement may be very small.

In the noisy setting, Kääriäinen showed a lower bound on the number of label queries required for any active learner to achieve a particular generalization error relative to the inherent noise rate [Kää06]. This lower bound is matched in certain cases by an algorithm developed by Balcan, Beygelzimer, and Langford called A^2 [BBL06] (which we discuss in

Chapter 2). A^2 was subsequently analyzed by Hanneke, who proved an upper bound on its label complexity in terms of a parameter called the disagreement coefficient [Han07]. The disagreement coefficient was further studied in [Fri09, Wan09], giving further justification to algorithms with label complexity bounded in terms of this quantity. Koltchinskii remarks that a similar parameter was previously used for studying ratio-type empirical processes, which has applications in passive learning [Ale87, GK06, Kol09]. A generalization of the disagreement coefficient to a certain class of loss functions was presented in [BDL09]. This work of [BDL09] also presents a framework called importance weighted active learning (which we discuss in Chapter 5), upon which one of the algorithms in this dissertation is based. (One of the algorithms in [BDL09] generalizes an algorithm developed in this dissertation in Chapter 4.) Finally, restrictions on the noise model (based on the low-noise condition of [Tsy04]) have also been studied and algorithmically exploited [BBZ07, CN06, CN07, Han09, Kol09]; under these restrictions, the achievable label complexity interpolates between what is achievable in the noise-free setting and the general (agnostic) noisy setting considered by [Kää06, BBL06].

1.4 Summary of Contributions

We begin by studying general approaches to active learning based on the idea of uncertainty sampling—querying the label of points about which the learner is "uncertain" in a precise sense. In Chapter 2, we describe the methods of [CAL94] and [BBL06] for (respectively) PAC and agnostic active learning. We describe a novel analysis of the [CAL94] procedure, specifically using a parameter called the *disagreement coefficient*, which was used by [Han07] for analyzing the [BBL06] procedure. This analysis turns out to be tighter than the corresponding analysis of [BBL06] when specialized to the PAC setting. We compare the results of both analyses to other upper and lower bounds for active and passive learning.

Both of the procedures from [CAL94] and [BBL06] are algorithmically underspecified: they require a mechanism for maintaining a subset of the hypotheses still under consideration by the algorithm; doing this efficiently was a challenge in the work of [CAL94] and not addressed by [BBL06]. In Chapters 3, 4, and 5, we present reduction-based active learning methods that are more algorithmic. In Chapter 3, we show how the procedure of [CAL94] can be viewed as a reduction to a very standard form of PAC learning. This immediately yields efficient procedures for PAC active learning whenever the corresponding PAC (passive) learning problems can be solved. The analysis also yields general upper bounds for these PAC active learning problems.

In Chapter 4, we show how to make the procedure of [BBL06] more algorithmic by recasting it using reductions to a particular form of agnostic learning. Unfortunately, this algorithm, like the [BBL06] procedure, suffers from a suboptimal analysis when specialized to the PAC setting. Thus, we also describe a more straightforward extension of the [CAL94] procedure that *does* recover the tighter analysis. We show how the corresponding notions used in the analysis of the [CAL94] procedure carry over to the agnostic setting for this new algorithm.

In Chapter 5, we describe two new algorithms based on reductions to simpler forms of agnostic learning. These algorithms have qualitative advantages over those in Chapter 4.

The first algorithm is a relaxation of the second method from Chapter 4 that allows for the use of reductions to simpler forms of agnostic learning. The second algorithm is based on importance weighting [BDL09], a technique for ensuring unbiased error estimates. We present a novel analysis of these error estimates, which is crucial for the analysis of the importance weighting active learning algorithm itself.

1.5 Learning Framework

We develop and analyze our algorithms with the standard PAC and agnostic learning frameworks in mind [Val84, KSS94]. These are standard in the study of supervised (passive) learning, and thus will allow us to compare the performance of our active learning procedures to their passive learning counterparts.

Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \mathcal{Y}$ where $\mathcal{X}$ is the input space and $\mathcal{Y} = \{\pm 1\}$ are the possible labels. Let $(X, Y) \in \mathcal{X} \times \mathcal{Y}$ be a pair of random variables with joint distribution $\mathcal{D}$. Here, X represents an unlabeled data point, and Y is its corresponding label.

Let $\mathcal{H}$ be a set of hypotheses mapping from $\mathcal{X}$ to $\mathcal{Y}$. The error of a hypothesis $h : \mathcal{X} \to \mathcal{Y}$ is

$$\operatorname{err}(h) \ := \ \Pr(h(X) \neq Y).$$

Let $h^* \ := \ \arg\min\{\operatorname{err}(h) : h \in \mathcal{H}\}$ be a hypothesis of minimum error in $\mathcal{H}$—we assume for simplicity that the minimum always exist. The goal of the learner is to return a hypothesis $h \in \mathcal{H}$ with error $\operatorname{err}(h)$ not much more than $\operatorname{err}(h^*)$.

In the realizable (PAC) setting of learning (active and passive), we assume that h^* has zero error $\operatorname{err}(h^*) = 0$, *i.e.*, that the labels perfectly correspond to the optimal hypothesis h^*. In this case, we can simply write $\operatorname{err}(h) = \Pr(h(X) \neq h^*(X)$, since the conditional distribution of Y given $X = x$ is deterministic. In the agnostic setting, the distribution of (X, Y) is arbitrary—no assumption is made about $\operatorname{err}(h^*)$.

We assume a learner has access to independent and identically distributed (iid) copies of the pair (X, Y). However, the active learner is not immediately given access to the labels; the labels are hidden from the learner unless the learner explicitly queries to see them. The active learner therefore has the added objective of minimizing the number of label queries (in addition to returning a low-error hypothesis).

The sample complexity of an algorithm (with respect to $\mathcal{D}$ and $\mathcal{H}$) is the required number of labeled examples randomly drawn from $\mathcal{D}$ so that, with probability at least $1 - \delta$ over the choice of the random examples, the algorithm produces a hypothesis $h \in \mathcal{H}$ with error $\operatorname{err}(h) \leq \operatorname{err}(h^*) + \epsilon$. The label complexity of an active learning algorithm is the number of label queries required to achieve the same goal. Note that a standard passive learning algorithm can be viewed as an active learning algorithm that simply queries every label. Therefore it has label complexity equal to its sample complexity. We are interested in active learning algorithms that improve on this baseline.

Chapter 2

Active Learning with Version Spaces

We present two active learning algorithms based on a simple version space approach, as well as a concept called the *disagreement coefficient* for analyzing the label complexity of active learning algorithms.

2.1 Introduction

One technique that has proved theoretically profitable is to maintain a candidate set of hypotheses (sometimes loosely called a *version space*), and to query the label of a point only if there is disagreement within this set about how to label the point. Note that if there is no disagreement within the set about how to label a point (*i.e.* every hypothesis there labels the point the same way), then the label of the point cannot be used to distinguish between any hypotheses in the set. Now, the criteria for membership in this candidate set need to be carefully defined so that the optimal hypothesis h^* is always included, but otherwise the set can be quickly pared down as more labels are queried.

To apply this technique, we need to resolve two issues: (i) what are the criteria for membership in the candidate set, and (ii) if there are several data points of disagreement for a candidate set, which one should we label?

In this chapter, we describe two algorithms based on this paradigm: the first for the (realizable) PAC setting due to [CAL94], and the second for the agnostic setting due to [BBL06]. The algorithms differ in the way they address the first issue above (due to the assumptions made in PAC learning), but are similar in the way they address the second.

2.2 PAC Active Learning

2.2.1 Algorithm

In PAC (active) learning, we assume there exists a hypothesis $h^* \in \mathcal{H}$ that correctly labels every example, *i.e.* $\Pr(h^*(X) = Y) = 1$. Although this is often an unrealistic assumption in practice, we will see that some of the algorithmic ideas in this setting can be transferred to the more realistic agnostic setting.

The algorithm of [CAL94], which we henceforth refer to as CAL, is shown in Figure 2.1. CAL proceeds by examining the unlabeled data points $X_1, X_2, \ldots$ one at a time, and decides after each point X_t whether or not to examine (*i.e.* query) its label Y_t. Whenever CAL doesn't query the label Y_t, it synthesizes one $\tilde{Y}_t$ on its own. Thus, after examining t unlabeled data points, the algorithm has a set of t labeled examples Z_t.

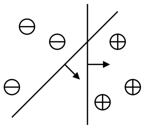
As suggested at the beginning of this chapter, the choice of whether or not to query y_t is made based on whether there is disagreement about how to label x_t among hypotheses in the *version space* $\mathcal{V}_t := \mathcal{V}(Z_t)$, where

Definition 2.1. *For a set of labeled examples $Z \subset \mathcal{X} \times \mathcal{Y}$, the* version space $\mathcal{V}(Z)$ with respect to a hypothesis class $\mathcal{H}$ *is*

$$\mathcal{V}(Z) := \{h \in \mathcal{H} : h(x) = y \ \forall (x, y) \in Z\}$$

the subset of hypotheses in $\mathcal{H}$ consistent with the examples in Z.

For example, suppose the hypothesis class $\mathcal{H}$ is the set of linear separators in the plane. The version space for the set of points depicted below contains all linear separators consistent with the labeling of these points.



(Just two of the hypotheses are depicted.)

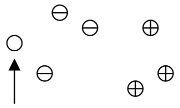
Formally, CAL chooses to query the label Y_t if and only if X_t is in the *region of disagreement* $\mathcal{R}(\mathcal{V}_{t-1})$ for $\mathcal{V}_{t-1}$:

Definition 2.2. *For a set of hypotheses V, the* region of disagreement[1] $\mathcal{R}(V)$ *is*

$$\mathcal{R}(V) := \{x \in \mathcal{X} : \exists h, h' \in V \text{ such that } h(x) \neq h'(x)\}$$

the set of unlabeled examples x for which there are hypotheses in V that disagree on how to label x.

Continuing our previous example, the indicated point in the figure below is *not* in the region of disagreement with respect to the current version space.



Clearly, all of the hypotheses in the version space also assign the indicated point a -1 label.

Note that Step 2(b) in CAL is unnecessary (and can simply be replaced by $Z_t := Z_{t-1}$). This is because if Step 2(b) is executed in, say, iteration t, then every $h \in \mathcal{V}_{t-1}$ has $h(X_t) = \tilde{Y}_t$; in this case,

$$\mathcal{V}_t = \mathcal{V}_{t-1} \cap \{h \in \mathcal{V}_{t-1} : h(X_t) = \tilde{Y}_t\} = \mathcal{V}_{t-1}.$$

Put another way, the version space is unchanged by the synthesized labels.

[1] [CAL94] defines a *region of uncertainty* with respect to a set of labeled examples $Z \subset \mathcal{X} \times \mathcal{Y}$; it is equivalent to $\mathcal{R}(\mathcal{V}(Z))$.

```
Algorithm 2.1 (CAL)
Initialize: $Z_0 := \emptyset$, $\mathcal{V}_0 := \mathcal{H}$.
For $t = 1, 2, \ldots, n$:
    1. Obtain unlabeled data point $X_t$.
    2. If there exist $h, h' \in \mathcal{V}_{t-1}$ such that $h(X_t) \neq h'(X_t)$:
        (a) Then: Query $Y_t$, and set $Z_t := Z_{t-1} \cup \{(X_t, Y_t)\}$.
        (b) Else: Set $\tilde{Y}_t := h(X_t)$ for any $h \in \mathcal{V}_{t-1}$, and set $Z_t := Z_{t-1} \cup \{(X_t, \tilde{Y}_t)\}$.
    3. Set $\mathcal{V}_t := \{h \in \mathcal{H} : h(X_i) = Y_i \ \forall (X_i, Y_i) \in Z_t\}$.
Return: any $h \in \mathcal{V}_n$.
```

Figure 2.1: The algorithm of [CAL94] for PAC active learning.

2.2.2 Correctness Analysis

CAL correctly deduces the labels assigned by h^* whenever it doesn't query the true label Y_t. This is formalized in the following theorem.

Theorem 2.1. *Assume $h^* \in \mathcal{H}$ satisfies $h^*(X_t) = Y_t$ for all t. Every label $\tilde{Y}_t$ synthesized by CAL (in Step 2(b)) is the true label (*i.e. *$\tilde{Y}_t = h^*(X_t)$).*

Proof. By induction on t. If CAL synthesizes the label $\tilde{Y}_t$, then every hypothesis $h \in \mathcal{V}_{t-1}$ assigns $h(X_t) = \tilde{Y}_t$. If $t = 1$ (the base case), then $h^* \in \mathcal{H} = \mathcal{V}_0$ so $h^*(X_t) = \tilde{Y}_t$. If $t > 1$, we assume as the inductive hypothesis that $Z_{t-1} = \{(X_1, h^*(X_1)), \ldots, (X_{t-1}, h^*(X_{t-1}))\}$; this implies $h^* \in \mathcal{V}_{t-1}$ so $h^*(X_t) = \tilde{Y}_t$. □

We conclude that the final hypothesis returned by CAL after seeing n random unlabeled examples is, in fact, consistent with n random *labeled* examples. This means that CAL has label complexity bounded by that of a passive learning algorithm that simply returns a hypothesis consistent with a given labeled sample.

2.2.3 Disagreement Coefficient

The cases in which CAL will have an improved label complexity over passive learning are particular to the hypothesis class $\mathcal{H}$ and the data distribution $\mathcal{D}$. The relevant quantity that characterizes the label complexity of CAL is the *disagreement coefficient*, which was used in [Han07] for analyzing the label complexity of the A^2 algorithm of [BBL06]. A similar quantity was previously used for studying ratio-type empirical processes in passive learning [Ale87, GK06, Kol09].

To introduce the disagreement coefficient, we first define a metric on the set of hypotheses $\mathcal{H}$.

Definition 2.3. *For a random variable $X \in \mathcal{X}$, the* disagreement (pseudo) metric *ρ on $\mathcal{H}$ is defined by*

$$\rho(h, h') := \Pr(h(X) \neq h'(X)).$$

The disagreement metric is a pseudo-metric because we may have $\rho(h, h') = 0$ but $h \neq h'$. Nevertheless, it inherits the other metric properties from the L_1 probability metric induced by the distribution of X (*e.g.*, the triangle inequality).

Let $B(h, r) := \{h' \in \mathcal{H} : \rho(h, h') \leq r\}$ denote the ball centered at $h \in \mathcal{H}$ of radius $r \geq 0$. We can now define the disagreement coefficient.

Definition 2.4. *The* disagreement coefficient *$\theta(\mathcal{H}, \mathcal{D})$ with respect to a hypothesis class $\mathcal{H}$ and distribution $\mathcal{D}$ is*

$$\theta(\mathcal{H}, \mathcal{D}) := \sup \left\{ \frac{\Pr(X \in \mathcal{R}(B(h^*, r)))}{r} : r > 0 \right\} \tag{2.1}$$

where h^* is a (particular) hypothesis of minimum error under $\mathcal{D}$.

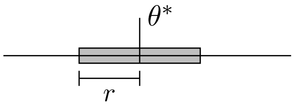
Note that we can also consider a variant of the disagreement coefficient in Eq. (2.1) so that the supremum is taken over $r = \Omega(\epsilon)$ where $\epsilon > 0$ is the target error rate. This is sensible because we may only care about the distinction between hypotheses up to some tolerable error rate ϵ. In this case, we always have $\theta(\mathcal{H}, \mathcal{D}) \leq O(1/\epsilon)$.

We now give some intuition behind the disagreement coefficient. Suppose in the course of active learning, an algorithm has narrowed down its current set of candidate hypotheses $\mathcal{V}_t$ to just those of error at most r_t. In the notation above, this means that

$$\mathrm{err}(h) \;=\; \Pr(h(X) \neq Y) \;=\; \Pr(h(X) \neq h^*(X)) \;=\; \rho(h, h^*) \;\leq\; r_t$$

for every $h \in \mathcal{V}_t$; *i.e.* $\mathcal{V}_t \subseteq B(h^*, r_t)$. Now, the only examples that can possibly help distinguish hypotheses in $\mathcal{V}_t$ are those in $\mathcal{R}(\mathcal{V}_t) \subseteq \mathcal{R}(B(h^*, r_t))$. This is because all $x \notin \mathcal{R}(\mathcal{V}_t)$ are labeled in the same way by every $h \in \mathcal{V}_t$. As learning progresses, we expect $\mathcal{V}_t$ to shrink and r_t to decrease: the algorithm will need to consider fewer hypotheses, and will be able to return a more accurate result. If the active learning algorithm simply chooses to query the label of any randomly chosen example that is in $\mathcal{R}(\mathcal{V}_t)$, then the size of this region relative to r_t will determine its label complexity. This is the ratio captured by the disagreement coefficient.

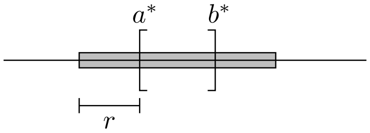
The disagreement coefficient is derived for various $(\mathcal{H}, \mathcal{D})$ pairs in [Han07]. If $\mathcal{H}$ is the class of single-variable threshold functions, and X has a uniform distribution on $[0, 1]$, then $\theta(\mathcal{H}, \mathcal{D}) = 2$. To see this, any $h_\theta \in B(h_{\theta^*}, r)$ has $\theta \in [\theta^* - r, \theta^* + r]$, which has probability mass $2r$.



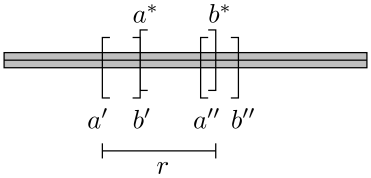
In the case of single-variable interval functions with the same distribution on X, we have that $\theta(\mathcal{H}, \mathcal{D}) = \max(4, 1/\Pr(h^*(X) = 1))$. To see this, note that if $r < (b^* - a^*)$, then any $h_{a,b} \in B(h_{a^*,b^*}, r)$ has $a \in [a^* - r, a^* + r]$ and $b \in [b^* - r, b^* + r]$, which accounts for a region of probability mass $4r$;

```
Algorithm 2.2 (Phased CAL)
Initialize: $Z_0 := \emptyset$, $\mathcal{V}_0 := \mathcal{H}$, $p := 0$, $t_0 := 0$.
For $t = 1, 2, \ldots, T$:
    1. Repeatedly sample $X_t$ until $X_t \in \mathcal{R}(\mathcal{V}_{t_p})$.
    2. Query $Y_t$, and set $Z_t := Z_{t-1} \cup \{(X_t, Y_t)\}$.
    3. Set $\mathcal{V}_t := \{h \in \mathcal{H} : h(X_i) = Y_i \ \forall (X_i, Y_i) \in Z_t\}$.
    4. If $\Pr(X \in \mathcal{R}(\mathcal{V}_t)) \leq \frac{1}{2}\Pr(X \in \mathcal{R}(\mathcal{V}_{t_p}))$, then set $t_{p+1} := t$ and $p := p + 1$
       (i.e. advance to the next phase).
Return: any $h \in \mathcal{V}_T$.
```

Figure 2.2: A variant of the CAL algorithm.



if $r \geq (b^* - a^*)$, then $B(h_{a^*,b^*}, r)$ contains every $h_{a,b}$ with $b - a \leq r - (b^* - a^*)$, and such hypotheses potentially disagree with h_{a^*,b^*} everywhere.



Finally, if $\mathcal{H} = \{h_w : w \in \mathbb{R}^d,\ h_w(x) = \operatorname{sgn}(w \cdot x)\}$ is the class of homogeneous linear separators in $\mathbb{R}^d$, and X is uniformly distributed on the surface of the unit ball $\{x \in \mathbb{R}^d : \|x\| = 1\}$, then $\pi\sqrt{d}/4 \leq \theta(\mathcal{H}, \mathcal{D}) \leq \pi\sqrt{d}$.

2.2.4 Label Complexity Analysis

We now give a label complexity analysis of CAL. The goal of this analysis is to determine the number of label queries required for the algorithm to return a hypothesis of error at most ϵ.

To more transparently illustrate the version space technique, we will actually analyze a slight variant of CAL which we call Phased CAL (Figure 2.2). Phased CAL has similar correctness and label complexity analyses as CAL proper.

The differences between Phased CAL and CAL proper are as follows. In Phased CAL, we assume for simplicity of analysis that $\mathcal{H}$ is finite. The arguments can be made to work for infinite classes using, say, covering arguments, but we omit these details because they distract from the core ideas.

Another difference is that the iterations of Phased CAL are divided into phases $p = 0, 1, 2, \ldots$. Let $I_p := \{t_p + 1, t_p + 2, \ldots, t_{p+1}\}$ be the iterations in phase p; each phase p is characterized by a version space $\mathcal{V}_{t_p}$ fixed at the beginning of the phase. By construction, the regions of disagreement $\mathcal{R}_{t_p} := \mathcal{R}(\mathcal{V}_{t_p})$ decrease geometrically in probability mass with p. Note that this aspect of Phased CAL is the main point of inefficiency relative to CAL proper. CAL will query Y_t only if X_t is in the region of disagreement for $\mathcal{V}_t$, which is just a subset of $\mathcal{V}_{t_p}$ if $t \in I_p$. Therefore the label complexity of Phased CAL essentially bounds the label complexity of CAL proper.

The final difference is that X_t for $t \in I_p$ is repeatedly sampled (*i.e.* independent copies are instantiated) until $X_t \in \mathcal{R}_{t_p}$. This can be seen as a form of rejection sampling—passing up unlabeled data points until we find one in $\mathcal{R}_{t_p}$. We note that this is actually only a cosmetic difference, since CAL proper effectively does this as well.

Theorem 2.2. *Fix any $\epsilon \in (0,1)$ and $\delta \in (0,1)$. With probability at least $1-\delta$, Phased CAL returns a hypothesis $h \in \mathcal{H}$ with $\mathrm{err}(h) \le \epsilon$ after querying at most*

$$O\left(\theta(\mathcal{H},\mathcal{D}) \cdot \left(\log \frac{|\mathcal{H}|}{\delta} + \log\log\frac{1}{\epsilon}\right) \cdot \log\frac{1}{\epsilon}\right)$$

labels.

Proof. Fix some $p \ge 0$. Let $\mathcal{H}_p := \{h \in \mathcal{V}_{t_p} : \rho(h, h^*) > r_p\}$ for some $r_p > 0$. These are the "bad" hypotheses in $\mathcal{V}_{t_p}$ that the algorithm will eliminate before moving onto the next phase. A bad hypothesis is eliminated (in Step 3) by observing an example (x, y) for which $h(x) \ne y$. For any $h \in \mathcal{H}_p$, we have

$$\Pr(h(X) \ne h^*(X) | X \in \mathcal{R}_{t_p}) \ge \frac{r_p}{\Pr(X \in \mathcal{R}_{t_p})} =: c_p$$

and thus

$$\Pr(h(X_t) = h^*(X_t)\ \forall t \in I_p | X_t \in \mathcal{R}_{t_p}\ \forall t \in I_p) \ \le\ (1 - c_p)^{t_{p+1} - t_p}$$

by the independence of the x_t. By a union bound over all $h \in \mathcal{H}_p$, we have

$$\Pr(\exists h \in \mathcal{H}_p \,.\, h(X_t) = h^*(X_t)\ \forall t \in I_p | X_t \in \mathcal{R}_{t_p}\ \forall t \in I_p) \le |\mathcal{V}_{t_p}|(1 - c_p)^{t_{p+1}-t_p} =: \delta_p.$$

The above inequality states that the probability the algorithm fails to eliminate all of the bad hypotheses $\mathcal{H}_p$ is at most δ_p.

In order to have $\Pr(X \in \mathcal{R}_{t_{p+1}}) \le (1/2)\Pr(X \in \mathcal{R}_{t_p})$, it suffices to ensure that $\mathcal{V}_{t_{p+1}} \subseteq B(h^*, r_p)$ with $r_p := \Pr(X \in \mathcal{R}_{t_p})/(2\theta(\mathcal{H},\mathcal{D}))$. This is because

$$\begin{aligned}
\Pr(X \in \mathcal{R}_{t_{p+1}}) &\le \Pr(X \in \mathcal{R}(B(h^*, r_p))) \\
&\le \theta(\mathcal{H},\mathcal{D}) \cdot r_p \\
&\le \theta(\mathcal{H},\mathcal{D}) \cdot \frac{\Pr(X \in \mathcal{R}_{t_p})}{2\theta(\mathcal{H},\mathcal{D})} \\
&= \frac{1}{2} \cdot \Pr(X \in \mathcal{R}_{t_p}).
\end{aligned}$$

With the above choice of r_p, we have $c_p = 1/(2\theta(\mathcal{H},\mathcal{D}))$ (for all p), so

$$\delta_p = |\mathcal{V}_{t_p}| \left(1 - \frac{1}{2\theta(\mathcal{H},\mathcal{D})}\right)^{t_{p+1}-t_p} \leq |\mathcal{V}_0| e^{-(t_{p+1}-t_p)/(2\theta(\mathcal{H},\mathcal{D}))}$$

using the fact $1 + a \leq e^a$ as well as the crude bound $|\mathcal{V}_{t_p}| \leq |\mathcal{V}_0|$. So, by a union bound over phases $p = 0, 1, \ldots, P-1$, the bad hypotheses $\mathcal{H}_p$ are eliminated in each phase p with probability at least

$$1 - |\mathcal{V}_0| \sum_{p=0}^{P-1} e^{-(t_{p+1}-t_p)/(2\theta(\mathcal{H},\mathcal{D}))}.$$

Note that if

$$t_{p+1} - t_p = \left\lceil 2\theta(\mathcal{H},\mathcal{D}) \log \frac{P|\mathcal{V}_0|}{\delta} \right\rceil$$

for each $0 \leq p \leq P$, then the success probability above is at least $1-\delta$. That is, with probability at least $1-\delta$, the algorithm will successfully eliminate the bad hypotheses $\mathcal{H}_p$ after querying at most $O(\theta(\mathcal{H},\mathcal{D}) \cdot \log(P|\mathcal{V}_0|/\delta))$ labels in each phase p. In this event, with $P = O(\log 1/\epsilon)$, the final hypothesis returned has error at most ϵ, and the number of labels queried is

$$\sum_{p=0}^{P-1} t_{p+1} - t_p = O\left(\theta(\mathcal{H},\mathcal{D}) \cdot \log\frac{1}{\epsilon} \cdot \left(\log\frac{|\mathcal{V}_0|}{\delta} + \log\log\frac{1}{\epsilon}\right)\right)$$

as claimed. □

When is the bound from Theorem 2.2 an improvement over passive learning? Suppose $\theta(\mathcal{H},\mathcal{D}) = O(1)$. The bound states that, in terms of ϵ, the number of labels required by Phased CAL is just $O(\log(1/\epsilon) \cdot \log\log(1/\epsilon))$. In contrast, the number of labeled examples required of a passive learner is $\Omega(1/\epsilon)$. Therefore, in this case, active learning provides an exponential improvement over passive learning in label complexity.

Of course, we have assumed here that $\theta(\mathcal{H},\mathcal{D})$ is bounded. which is not always the case. However, if we consider the alternative definition of $\theta(\mathcal{H},\mathcal{D})$ in which the supremum is taken over $r = \Omega(\epsilon)$ in Eq. (2.1), then it may be possible to explicitly consider the dependence on ϵ and achieve a tighter label complexity analysis. We may then view the analysis in Theorem 2.2 as applicable in the regime where $\theta(\mathcal{H},\mathcal{D})$ is constant, understanding that another analysis may better characterize what happens outside of this regime.

2.2.5 Discussion

We remark that CAL (and Phased CAL) is a suboptimal strategy for active learning for a simple reason: it is content with querying the label of *any* point in the disagreement region. This is evident in the analysis of Phased CAL, which says that the number of queries to advance from one phase to the next is roughly proportional to the disagreement coefficient $\theta(\mathcal{H},\mathcal{D})$. A better strategy would be to query a point that potentially eliminates as many bad hypotheses as possible; this may be many more than can be eliminated by a typical (random) point in the region of disagreement.

A simple example of this disparity is the case where X is uniformly distributed on the unit sphere in $\mathbb{R}^d$, and $\mathcal{H}$ is the set of homogeneous linear separators. In this case, $\theta(\mathcal{H}, \mathcal{D}) \approx \sqrt{d}$ [Han07]; however, there are always points to query that will eliminate a constant fraction of the bad hypotheses [Das05, BBZ07]. Therefore, CAL is roughly $\sqrt{d}$ times suboptimal in label complexity, due to its conservativeness in choosing points to query.

A more aggressive active learning strategy was discovered by Dasgupta, who characterized the label complexity of the algorithm by a sharper quantity called the *splitting index* [Das05]. However, Dasgupta's algorithm is computationally intractable, whereas CAL can be made computationally tractable, as we will see in the next chapter. We leave as an open problem whether Dasgupta's algorithm can be made computationally tractable, perhaps in the similar manner as CAL. Note that in special cases, there are efficient algorithms that achieve the same label complexity as Dasgupta's algorithm [FSST97, GBNT05, DKM05, BBZ07].

2.3 Agnostic Active Learning

Agnostic (active) learning differs from PAC (active) learning in that we no longer assume there exists a hypothesis in $\mathcal{H}$ that correctly labels every example. Therefore, the learner only hopes to return a hypothesis with error not much more than that of a hypothesis $h^* \in \mathcal{H}$ with minimum error

$$h^* := \arg\min_{h \in \mathcal{H}} \mathrm{err}_{\mathcal{D}}(h)$$

(we assume the minimum exists for simplicity).

The correctness of the CAL algorithm crucially relies on the assumption that the hypothesis class $\mathcal{H}$ contains a classifier h^* that perfectly labels all of the data. Such an assumption is both often unrealistic and potentially dangerous in practice. The second point here deserves further explanation. The consistency analysis in Theorem 2.1 actually implies that CAL is always able to find a hypothesis in $\mathcal{H}$ consistent with *its* set of labeled examples Z_t, even if no $h \in \mathcal{H}$ is consistent with the true labels. This is because the synthesized labels $\tilde{Y}_t$ always correspond to some $h \in \mathcal{V}_{t-1} \subseteq \mathcal{H}$, and true labels Y_t are only queried if there are hypotheses in $\mathcal{V}_{t-1}$ consistent with both $Y_t = +1$ and $Y_t = -1$. Therefore, CAL never discovers if the data is actually inconsistent with every hypotheses in $\mathcal{H}$ (which is often easily checked in the non-active setting), and the hypothesis returned can be significantly worse than the best hypothesis in the class. Thus assuming the existence of a perfect hypothesis can be a self-fulfilling delusion in active learning. Related inconsistency issues arise with a variety of active learning methods, including many based on uncertainty sampling and other similar heuristics.

2.3.1 Algorithm

The first algorithm designed for the agnostic setting is the A^2 algorithm of [BBL06], specified in Figure 2.3; it can be seen as a generalization of CAL (or Phased CAL) to the agnostic setting. A^2 proceeds in phases $p = 0, 1, 2, \ldots$ relative to version spaces $\mathcal{V}_p$, and points are drawn from the distribution conditioned falling in $\mathcal{R}(\mathcal{V}_p)$. The variable k is used to index a sequence of confidence parameters $\delta_k := \delta/(k^2 + k)$, which guarantees that $\sum_{k \geq 1} \delta_k \leq \delta$.

```
**Algorithm 2.3** ($A^2$)
Notes: $\delta_k := \delta/(k^2+k)$ for all $k \geq 1$.
Initialize: $Z_0 := \emptyset$, $\mathcal{V}_0 := \mathcal{H}$, $k := 1$.
For $p = 0, 1, 2, \ldots$ until
    $$\Pr(X \in \mathcal{R}(\mathcal{V}_p)) \cdot \left( \min_{h' \in \mathcal{V}_p} \mathrm{UB}(Z_p, h', \delta_k) - \min_{h' \in \mathcal{V}_p} \mathrm{LB}(Z_p, h', \delta_k) \right) \leq \epsilon :$$
    1. Let $Z_p := \emptyset$, $n_p := 0$, $\mathcal{V}'_{p+1} := \mathcal{V}_p$, $k := k+1$.
    2. Repeat until $\Pr(X \in \mathcal{R}(\mathcal{V}'_{p+1})) \leq \frac{1}{2} \Pr(X \in \mathcal{R}(\mathcal{V}_p))$:
            If
                $$\Pr(X \in \mathcal{R}(\mathcal{V}_p)) \cdot \left( \min_{h' \in \mathcal{V}_p} \mathrm{UB}(Z_p, h', \delta_k) - \min_{h' \in \mathcal{V}_p} \mathrm{LB}(Z_p, h', \delta_k) \right) \leq \epsilon$$
            Then: return $h := \arg\min_{h' \in \mathcal{V}_p} \mathrm{UB}(Z_p, h', \delta_k)$.
            Else:
                i. Let $k := k+1$.
                ii. Let $n_p := 2n_p + 1$.
                iii. For $t = 1, \ldots, n_p$:
                    A. Repeatedly sample $X_{p,t}$ until $X_{p,t} \in \mathcal{R}(\mathcal{V}_p)$.
                    B. Query $Y_{p,t}$, and add $(X_{p,t}, Y_{p,t})$ to $Z_p$.
                iv. Let $\mathcal{V}'_{p+1} := \{h \in \mathcal{V}_p : \mathrm{LB}(Z_p, h, \delta_k) \leq \min_{h' \in \mathcal{V}_p} \mathrm{UB}(Z_p, h', \delta_k)\}$ and $k :=$
                    $k+1$.
    3. Let $\mathcal{V}_{p+1} := \mathcal{V}'_{p+1}$ and $p := p+1$.
Return: $h := \arg\min_{h \in \mathcal{V}_p} \mathrm{UB}(Z_p, h, \delta_k)$.
```

Figure 2.3: The A^2 algorithm of [BBL06] for agnostic active learning.

The core of the algorithm is in the "else" clause of Step 2: the algorithm draws and labels examples from the distribution $\mathcal{D}$ restricted to $\mathcal{R}(\mathcal{V}_p)$, and then eliminates hypotheses from $\mathcal{V}_p$ based on these examples. The aim here is to eliminate enough hypotheses so that the next disagreement region $\mathcal{R}(\mathcal{V}'_{p+1})$ is half as large in probability mass as the current one. This elimination step is based on error upper- and lower-bounds UB and LB. The requirement of these bounds is the following. If Z is a sample drawn iid from a distribution $\mathcal{D}$, then with probability at least $1-\delta$,

$$\mathrm{LB}(Z,h,\delta) \;\leq\; \mathrm{err}_{\mathcal{D}}(h) \;\leq\; \mathrm{UB}(Z,h,\delta)$$

for all $h \in \mathcal{H}$. For concreteness, we use

$$\mathrm{UB}(Z,h,\delta) := \mathrm{err}(h,Z) + O\left(\sqrt{\frac{d+\log(1/\delta)}{|Z|}}\right)$$

and

$$\mathrm{LB}(Z,h,\delta) := \mathrm{err}(h,Z) - O\left(\sqrt{\frac{d+\log(1/\delta)}{|Z|}}\right)$$

where d is the VC dimension of $\mathcal{H}$ [Tal94]. Because these bounds are allowed to fail with probability δ, the algorithm splits the overall allowed failure probability over successive applications of the bound by setting $\delta_k := \delta/(k^2+k)$.

Assuming the validity of these bounds, the algorithm ensures that the optimal hypothesis h^* is never eliminated. This is clear to see when the bounds are applied to errors with respect to $\mathcal{D}$: if a hypothesis h is eliminated, then $\mathrm{err}_{\mathcal{D}}(h) \geq \mathrm{LB}(Z_0,h,\delta_k) > \min_{h'} \mathrm{UB}(Z_0,h',\delta_k) \geq \mathrm{err}_{\mathcal{D}}(h^*)$, so $h \neq h^*$. The same logic also continues to hold by induction when applied to the restricted distributions. In this way, the algorithm pares down the version space while ensuring convergence towards h^*.

2.3.2 Label Complexity Analysis

The following theorem about A^2 is due to [Han07].

Theorem 2.3 ([Han07]). *With probability at least $1-\delta$, A^2 returns a hypothesis h with error $\mathrm{err}(h) \leq \mathrm{err}(h^*) + \epsilon$ and requests at most*

$$O\left(\theta(\mathcal{H},\mathcal{D})^2 \cdot \left(\frac{\mathrm{err}(h^*)^2}{\epsilon^2} + 1\right) \cdot \left(d\log^2\frac{1}{\epsilon} + \left(\log\frac{1}{\delta} + \log\log\frac{1}{\epsilon}\right) \cdot \log\frac{1}{\epsilon}\right)\right)$$

labels, where d is the VC dimension of $\mathcal{H}$.

To interpret the label complexity guarantee, we first ignore the dependence on the disagreement coefficient. The *supervised* sample complexity is

$$O\left(\left(\frac{\mathrm{err}(h^*)}{\epsilon^2} + \frac{1}{\epsilon}\right) \cdot \left(d\log\frac{1}{\epsilon} + \log\frac{1}{\delta}\right)\right)$$

[VC71]. Up to logarithmic factors, the $d\,\mathrm{err}(h^*)/\epsilon^2$ term in the supervised bound is scaled by a factor of $\mathrm{err}(h^*)$ in the A^2 bound, while the explicit d/ϵ term in the supervised bound is reduced to d. Note that if $\mathrm{err}(h^*) = 0$, then the label complexity is completely reduced to d times logarithmic factors in $1/\epsilon$.

We can also compare this upper bound for A^2 to lower bounds on the number of labels that *any* active learner must query in order to produce a hypothesis of error at most $\mathrm{err}(h^*) + \epsilon$. First, there is a lower bound of

$$\Omega\left(\frac{d\,\mathrm{err}(h^*)^2}{\epsilon^2}\right)$$

due to [BDL09]. This explains the $d\,\mathrm{err}(h^*)^2/\epsilon^2$ term in the A^2 bound.

Second, there is an information-theoretic lower bound of

$$\Omega(\log \mathcal{M}(\epsilon))$$

where $\mathcal{M}(\epsilon)$ is the size of the largest ϵ-packing of the metric space $(\mathcal{H}, \rho)$ (this is argued in [KMT93]: each label query provides a single bit of information, and at least $\log \mathcal{M}(\epsilon)$ bits are needed to describe a hypothesis in a maximal packing). A result due to [Hau95] states that there exist data distributions for which $\mathcal{M}(\epsilon) = \Omega((1/\epsilon)^d)$. Therefore, we have a lower bound of

$$\Omega(d \log 1/\epsilon).$$

This explains the d term in the A^2 bound.

Now we consider the disagreement coefficient. If $\mathrm{err}(h^*) = 0$, then Phased CAL depends only linearly on $\theta(\mathcal{H}, \mathcal{D})$, whereas A^2 depends quadratically on it. In fact, this is not due to slack in the analysis; it is shown in [Han07] that A^2 queries at least $\Omega(\theta(\mathcal{H}, \mathcal{D})^2)$ labels. It was posed as an open question by Hanneke whether this *quadratic* dependence was necessary of any agnostic active learner. We will see in a later chapter that, in fact, a *linear* dependence is sufficient. It is, however, an open question as to whether *any* dependence on the disagreement coefficient is necessary in the label complexity of all active learners.

Chapter 3

Reduction to PAC Learning

We recast the active learning algorithm of [CAL94] as a reduction to PAC learning. This view of the algorithm also leads to a simpler analysis of its label complexity.

3.1 Introduction

The CAL algorithm from the previous chapter (Algorithm 2.1) may appear to require explicit bookkeeping of which hypotheses remain in the version space $\mathcal{V}_t$. While this is certainly doable when the hypothesis class is small and easily enumerable, it appears intractable for large or infinite classes.

However, with some thought, it can be seen that the task of determining membership in the version space $\mathcal{V}_t$ can be reduced to a standard method for PAC learning – that of checking the existence of a hypothesis consistent with a set of labeled data. This can be much easier than an explicit enumeration of the version space. For instance, when $\mathcal{H}$ is the class of linear separators in $\mathbb{R}^d$, checking for the existence of consistent hypothesis can be done by solving a simple linear program.

Viewing CAL as a reduction not only provides a tractable implementation for many hypothesis classes, it also allows for a simpler label complexity analysis. The analysis given in Theorem 2.2 focuses on eliminating hypotheses in the version space $\mathcal{V}_t$. We will show another, comparable analysis that avoids this fixation on $\mathcal{V}_t$. Instead, the analysis will more directly relate to the task of finding consistent hypotheses, *i.e.*, the reduction to PAC learning.

3.2 A Reduction-based Characterization of CAL

The key to the reduction (Algorithm 3.1) is that the version space $\mathcal{V}_t$ used by CAL can be implicitly tracked through the labeled sample Z_t. The existence of $h, h' \in \mathcal{V}_{t-1}$ such that $h(X_t) \neq h'(X_t)$ is equivalent to the existence of both

1. a hypothesis $h^{+1} \in \mathcal{H}$ consistent with $Z_{t-1} \cup \{(X_t, +1)\}$, and

2. a hypothesis $h^{-1} \in \mathcal{H}$ consistent with $Z_{t-1} \cup \{(X_t, -1)\}$.

```
Algorithm 3.1 (Reduction-based CAL)
Initialize: $Z_0 := \emptyset$.
For $t = 1, 2, \ldots, n$:
    1. Obtained unlabeled data point $X_t$.
    2. If there exists both
        • $h^{+1} \in \mathcal{H}$ consistent with $Z_{t-1} \cup \{(X_t, +1)\}$, and
        • $h^{-1} \in \mathcal{H}$ consistent with $Z_{t-1} \cup \{(X_t, -1)\}$
       (a) Then: Query $Y_t$, and set $Z_t := Z_{t-1} \cup \{(X_t, Y_t)\}$.
       (b) Else if only $h^y$ exists (for some $y \in \{\pm 1\}$): Set $\tilde{Y}_t := y$ and set $Z_t := Z_{t-1} \cup \{(X_t, \tilde{Y}_t)\}$.
Return: any $h \in \mathcal{H}$ consistent with $Z_n$.
```

Figure 3.1: The CAL algorithm recast using reductions.

Finding hypotheses consistent with a set of labeled examples is a standard approach to PAC learning. Indeed, it is well-known that if Z_n is a random sample of n examples labeled by some $h^* \in \mathcal{H}$, then with probability at least $1 - \delta$, any $h \in \mathcal{H}$ consistent with Z_n has error $\mathrm{err}(h)$ at most

$$O\left(\frac{1}{n}\left(d \log n + \log \frac{1}{\delta}\right)\right) \tag{3.1}$$

where d is the VC dimension of $\mathcal{H}$ [BEHW89]. Moreover, the task of finding consistent hypotheses can be reduced to (distribution-free) PAC learning. In this sense, CAL can be viewed as a reduction from PAC active learning to PAC supervised learning.

To see the claimed equivalence of Algorithm 2.1 (CAL) and Algorithm 3.1 (Reduction-based CAL), we proceed by induction as in Theorem 2.1 by assuming $h^*(X) = Y$ for all $(X, Y) \in Z_{t-1}$, including the synthesized labels. This means $h^* \in \mathcal{V}_{t-1}$. If both h^{+1} and h^{-1} exist, then they are both consistent with Z_{t-1} and therefore both in $\mathcal{V}_{t-1}$. But because $h^{+1}(X_t) \neq h^{-1}(X_t)$, CAL would query the label $Y_t = h^*(X_t)$ in this case. On the other hand, if (say) h^{-1} does not exist, then no hypothesis consistent with Z_{t-1} is also consistent with $(X_t, -1)$. This means every hypothesis consistent with Z_{t-1} (*i.e.*, every hypothesis in $\mathcal{V}_{t-1}$), including h^*, must label X_t as $+1$.

3.3 A Simpler Label Complexity Analysis

In light of the reduction in Algorithm 3.1, we can give a simpler label complexity analysis of the algorithm.

3.3.1 Some Refined Disagreement Metric Notions

First, we define some refined notions of *region of disagreement* and the *disagreement coefficient.*

Definition 3.1. *The* region of disagreement $\mathcal{R}(h, r)$ *of radius* r *around a hypothesis* $h \in \mathcal{H}$ *in the disagreement metric space* $(\mathcal{H}, \rho)$ *is*

$$\mathcal{R}(h, r) := \{x \in \mathcal{X} : \exists h' \in B(h, r) \text{ such that } h(x) \neq h'(x)\}$$

the set of unlabeled examples x *for which there exists a hypothesis* h' *at distance at most* r *from* h *(under* ρ*) that disagrees with* h *on* x*.*

The quantity is more refined than the earlier notion of the region of disagreement with respect to a set of hypotheses V, because it restricts attention to disagreement with a particular hypothesis, rather than any pair of hypotheses V. For instance, we have $\mathcal{R}(h^*, r) \subseteq \mathcal{R}(B(h^*, r))$, but the reverse may not be true.

Definition 3.2. *The* disagreement coefficient $\theta(h, \mathcal{H}, \mathcal{D})$ *for a hypothesis* $h \in \mathcal{H}$ *in the disagreement metric space* $(\mathcal{H}, \rho)$ *is*

$$\theta(h, \mathcal{H}, \mathcal{D}) := \sup \left\{ \frac{\Pr(X \in \mathcal{R}(h, r))}{r} : r > 0 \right\}.$$

We will often simply write θ to mean $\theta(h^*, \mathcal{H}, \mathcal{D})$.

3.3.2 Label Complexity Analysis

We are now ready to give the new label complexity analysis.

Theorem 3.1. *Conditioned on an event that occurs with probability at least* $1 - \delta$*, the expected number of labels queried by Reduction-based CAL after* n *iterations is at most*

$$O\left(\theta \cdot \left(d \log n + \log \frac{1}{\delta}\right) \cdot \log n\right).$$

Proof. First, we argue that with probability at least $1 - \delta$, we have the following property. For all $t \geq 1$, every $h \in \mathcal{H}$ consistent with Z_t has error $\operatorname{err}(h)$ at most

$$O\left(\frac{1}{t}\left(d \log t + \log \frac{t(t+1)}{\delta}\right)\right). \tag{3.2}$$

This simply follows from applying Eq. (3.1) for every $t \geq 1$, substituting $\delta/(t^2 + t)$ in place of δ, and then applying a union bound over all t.

Now we condition on the event that the above property holds. The algorithm queries Y_t if and only if there exists $h \in \mathcal{H}$ such that:

1. h is consistent with Z_{t-1}, and

2. h disagrees with h^* on X_t.

The first condition implies that the error of such a hypothesis $h \in \mathcal{H}$ can be bounded using Eq. (3.2), *i.e.* $\mathrm{err}(h) \leq O((1/t)(d \log t + \log(t/\delta)))$. Let r_t denote the value of this bound. This, combined with the second condition, implies that X_t is in the region of disagreement for the subset of hypotheses at most r_t away from h^*, *i.e.* $X_t \in \mathcal{R}(h^*, r_t)$. By the definition of the disagreement coefficient θ, we have

$$\Pr(X_t \in \mathcal{R}(h^*, r_t)) \leq \theta \cdot r_t.$$

Let $Q_t \in \{0, 1\}$ be the random variable that indicates if the label Y_t is queried. Then the expected number of queries after n iterations is

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{n} Q_t\right] &= \sum_{t=1}^{n} \mathbb{E}[\mathbb{E}[Q_t | Z_{t-1}, X_t]] \\
&\leq \sum_{t=1}^{n} \mathbb{E}\left[\Pr(\exists h \in \mathcal{H} \text{ consistent with } Z_{t-1} \cup \{(X_t, -h^*(X_t))\} | Z_{t-1}, X_t)\right] \\
&\leq \sum_{t=1}^{n} \Pr(\exists h \in B(h^*, r_t) \text{ s.t. } h(X_t) \neq h^*(X_t)) \\
&= \sum_{t=1}^{n} \Pr(X_t \in \mathcal{R}(h^*, r_t)) \\
&\leq \sum_{t=1}^{n} \theta \cdot O\left(\frac{1}{t}\left(d \log t + \log \frac{t(t+1)}{\delta}\right)\right) \\
&= O\left(\theta \cdot \left(d \log n + \log \frac{1}{\delta}\right) \cdot \log n\right)
\end{aligned}$$

as claimed. □

The label complexity guarantee in Theorem 3.1 can be related back to that in Theorem 2.2 as follows. First, while Theorem 3.1 is stated in terms of the (conditional) expectation of the number of labels queried, it can easily be converted into a high-probability guarantee by simply applying standard large deviation bounds for martingales (see, *e.g.*, [MR95]). Second, to give a label complexity guarantee in terms of the target error rate ϵ, one simply needs to substitute a value of n for which the error in Eq. (3.1) is at most ϵ. This is because the final hypothesis returned by the algorithm is consistent with n labeled examples drawn iid from $\mathcal{D}$, and therefore has error bounded as in Eq. (3.1). Finally, we allow any hypothesis class $\mathcal{H}$ with finite VC dimension d in Theorem 3.1 (but note that $d \leq \log |\mathcal{H}|$, so the bound here can only be tighter) and use the more refined variant of the disagreement coefficient[1]. Now, it should be clear that the two stated theorems provide essentially the same guarantee on the label complexity of CAL.

[1] We note that the argument in Theorem 2.2 can be reworked in terms of the VC dimension d by working with a finite covering of $\mathcal{H}$ of size $O((1/\epsilon)^d)$, and also be written in terms of $\theta(h^*, \mathcal{H}, \mathcal{D})$ rather than $\theta(\mathcal{H}, \mathcal{D})$.

Chapter 4

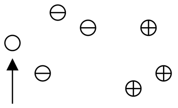
Reduction to Agnostic Learning I: Implicit Version Spaces

We recast the A^2 algorithm of [BBL06] as a reduction to a form of agnostic learning. We also describe a generalization the algorithm of [CAL94] to the agnostic setting.

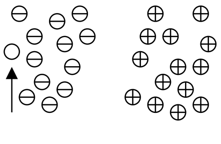
4.1 Introduction

Viewing the CAL algorithm using reductions to a form of PAC learning is rather straightforward, as testing for membership in the current version space precisely corresponds to checking for consistency with the current set of labeled examples. Such a test fails in the agnostic setting since it may be that no hypothesis in the version space is consistent with the examples.

Recall our example from Chapter 2 where $\mathcal{H}$ was the class of two-dimensional linear separators, and the first six data were labeled in the following manner.



If it is assumed that the data is linearly separable (*i.e.*, the labels correspond to some $h^* \in \mathcal{H}$), then the label of the indicated point is already determined: it must be labeled -1, as every linear separator consistent with the first six data label it as -1. But this logic is invalid without the separability assumption, as it could be that the optimal hypothesis in $\mathcal{H}$ disagrees with these first six labels. The sampling error with just six data is too large to reliably conclude that the optimal hypothesis would label the seventh point as -1. On the other hand, with more labeled data (as depicted below), such a conclusion becomes more plausible.



A hypothesis that labels the indicated point +1 label would have to misclassify many other points—an unlikely event if the optimal hypothesis has low error.

We need to generalize the test developed for the separable setting to the agnostic setting. Our reduction will divide the examples into two sets, $\tilde{S}$ and T (the significance of the ornament on $\tilde{S}$ will be explained later). The set $\tilde{S}$ will always contain examples consistent with the best hypothesis in the class, while T may contain examples on which even the best hypothesis errs. Assuming we can manage this, it would be reasonable for a learning algorithm to simply locate a hypothesis $h \in \mathcal{H}$ consistent with $\tilde{S}$, and otherwise with minimum error with respect to T. Thus, the form of agnostic learning we use for our reductions is a variant of empirical risk minimization, which we encapsulate in the following subroutine LEARN$_{\mathcal{H}}$:

> LEARN$_{\mathcal{H}}(\tilde{S}, T)$ returns $h \in \mathcal{H}$ such that $\mathrm{err}(h, \tilde{S}) = 0$ and $\mathrm{err}(h, T)$ is minimum over all $h \in \mathcal{H}$. If no hypothesis $h \in \mathcal{H}$ is consistent with $\tilde{S}$, signal this failure by returning $\perp$.

It is well-known that standard empirical risk minimization is a consistent method for agnostic learning: if Z_t is a random sample of t examples from $\mathcal{D}$, then with probability at least $1-\delta$, the empirical minimizer $h := \arg\min_{h\in\mathcal{H}} \mathrm{err}(h, Z_n)$ has error $\mathrm{err}(h)$ at most

$$\mathrm{err}(h^*) + O\left(\sqrt{\mathrm{err}(h^*) \cdot \frac{d \log n + \log(1/\delta)}{n}} + \frac{d \log n + \log(1/\delta)}{n}\right) \tag{4.1}$$

where d is the VC dimension of $\mathcal{H}$ and $h^* \in \mathcal{H}$ is a hypothesis of minimum (true) error [VC71]. Therefore, the extra stipulation—requiring the hypothesis $h :=$ LEARN$_{\mathcal{H}}(\tilde{S}, T)$ be consistent with $\tilde{S}$—is the mechanism we use to maintain an *implicit version space*, which we hope to gradually reduce in the same manner as the CAL algorithm for the PAC setting.

We note that exact implementation of the LEARN$_{\mathcal{H}}$ subroutine is often intractable in high dimensions. Indeed, agnostic supervised learning is computationally hard for many hypothesis classes such as half-spaces [Fel06, GR06], and of course, agnostic active learning is at least as hard in the worst case. However, we will see that the LEARN$_{\mathcal{H}}$ subroutine is only called on samples from the underlying unlabeled data distribution, and not on pathologically hard instances (like those arising from hardness reductions) unless they are inherent in the data. Therefore, we think of LEARN$_{\mathcal{H}}$ as an ideal abstraction of agnostic supervised learning, with the understanding that it may be only approximately implemented in practice.

4.2 A Reduction-based Variant of A^2

We first describe a method for recasting the A^2 algorithm of [BBL06] using reductions to agnostic learning. The reduction (Algorithm 4.1) is specified in terms of the LEARN$_{\mathcal{H}}$ subroutine detailed above. There are two key differences between Reduction-based A^2 algorithm and A^2 proper (Algorithm 2.3). First, Reduction-based A^2 operates on an initial sample U_0 of iid copies of X, whereas A^2 involves computing probabilities with respect to the distribution of X. Using a uniform distribution over the sample U_0 is sufficient as long as U_0 is large enough for true errors of hypotheses $\mathrm{err}(h)$ to be closely approximated by empirical errors computed with respect to the sample U_0 (assuming that the proper labels are given).

Algorithm 4.1 (Reduction-based A^2)

```
Notes: $\delta_k := \delta/(k^2 + k)$ for all $k \geq 1$; see Eq. (4.2) for the definition of $\Delta$.
Initialize: $U_0 := \{x_1, x_2, \ldots, x_m\}$, $\tilde{S}_0 := \emptyset$, $n_0 := 1$, $k := 0$.
For phase $p = 0, 1, 2, \ldots$:
   1. $k := k + 1$; $p_k := p$.
   2. Let $U'$ be a random subset of $U_p$ of size $n_p$.
   3. Let $T_p := \{(x_i, y_i) : x_i \in U'\}$, querying the labels $y_i$ as needed.
   4. Let $h_p := \text{LEARN}_{\mathcal{H}}(\tilde{S}_p, T_p)$.
   5. If $(|U_p|/|U_0|) \cdot \Delta(T_p, \delta_k) \leq \epsilon$, then return $h_p$.
   6. Let $U' := \emptyset$ and $\tilde{S}' := \emptyset$.
   7. For each $x \in U_p$:
      (a) Let $h'_{x,p} := \text{LEARN}_{\mathcal{H}}(\tilde{S}_p \cup \{(x, -h_p(x))\}, T_p)$.
      (b) If $h'_{x,p} = \perp$ or $\text{err}(h'_{x,p}, T_p) - \text{err}(h_p, T_p) > \Delta(T_p, \delta_k)$, then $U' := U' \cup \{x\}$ and
          $\tilde{S}' := \{(x, h_p(x))\}$.
   8. Let $U_{p+1} := U_p \setminus U'$ and $\tilde{S}_{p+1} := \tilde{S}_p \cup \tilde{S}'$.
   9. If $|U_{p+1}|/|U_0| \leq \epsilon$, then return $h_f := \text{LEARN}_{\mathcal{H}}(\tilde{S}_{p+1}, \emptyset)$.
  10. If $|U_{p+1}| > (1/2)|U_p|$:
      (a) Then: repeat phase $p$ with $n_p := 2n_p$.
      (b) Else: continue to phase $p + 1$ with $n_{p+1} := 1$.
```

Figure 4.1: The A^2 algorithm recast using reductions.

Second, Reduction-based A^2 implicitly maintains a version space through example-based constraints (enforced using the constraint mechanism in LEARN$_{\mathcal{H}}$). Examples are added to the constraint set $\tilde{S}$ only if they are consistent with the h^*, the best hypothesis in the class. This guarantees that h^* is never eliminated from the version space.

4.2.1 Active Learning with a Fixed Sample

We will view U_0 as the unlabeled components of m iid copies of (X, Y)—call them $(X_1, Y_1), \ldots, (X_m, Y_m)$. The labels Y_i corresponding to the X_i are hidden unless the algorithm queries for them. Let $Z_0 := \{(X_1, Y_1), \ldots, (X_m, Y_m)\}$ be the fully-labeled set. Uniform convergence results imply that

$$m = \Omega\left(\frac{1}{\varepsilon^2}\left(d + \log\frac{1}{\eta}\right)\right)$$

suffices to ensure, with probability at least $1 - \eta/2$, that $|\operatorname{err}(h) - \operatorname{err}(h, Z_0)| \leq \varepsilon/3$ for all $h \in \mathcal{H}$, where d is the VC dimension of $\mathcal{H}$ [Tal94]. If the goal of the learning algorithm is to return a hypothesis of error $\min_{h\in\mathcal{H}} \operatorname{err}(h) + \varepsilon$ with probability at least $1 - \eta$, we simply need to find a hypothesis of error at most $\min_{h\in\mathcal{H}} \operatorname{err}(h, Z_0) + \varepsilon/3$ with probability at least $1 - \eta/2$ (conditioned on the initial sample Z_0). This see that this is sufficient, let $h^*_{\mathcal{D}} := \arg\min_{h\in\mathcal{H}} \operatorname{err}(h)$ and $h^* := \arg\min_{h\in\mathcal{H}} \operatorname{err}(h, Z_0)$. If $\operatorname{err}(h, Z_0) \leq \operatorname{err}(h^*, Z_0) + \varepsilon/3$, then

$$\begin{aligned}\operatorname{err}(h) &= \operatorname{err}(h^*_{\mathcal{D}}) + (\operatorname{err}(h, Z_0) - \operatorname{err}(h^*_{\mathcal{D}}, Z_0)) \\ &\quad + (\operatorname{err}(h) - \operatorname{err}(h, Z_0)) + (\operatorname{err}(h^*_{\mathcal{D}}, Z_0) - \operatorname{err}(h^*_{\mathcal{D}})) \\ &\leq \operatorname{err}(h^*_{\mathcal{D}}) + (\operatorname{err}(h^*, Z_0) + \varepsilon/3 - \operatorname{err}(h^*_{\mathcal{D}}, Z_0)) + \varepsilon/3 + \varepsilon/3 \\ &\leq \operatorname{err}(h^*_{\mathcal{D}}) + \varepsilon.\end{aligned}$$

Therefore, we simply set the target $\epsilon := \varepsilon/3$ and allow probability of failure $\delta := \eta/2$; now treat the uniform distribution over a realization of the initial sample $Z_0 = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ as the base distribution for which h^* is optimal.

4.2.2 Deviation Bounds for Bootstrap Samples

As in the original A^2 algorithm, we will evaluate upper- and lower-bounds on the error of hypotheses. These bounds will be probabilistic, holding with probability at least $1-\delta$ over a random sample from Z_0 (or some subset thereof). Because we draw multiple random samples over the course of the algorithm, we allow the bounds to fail with probability $\delta_k := \delta/(k^2+k)$ on the kth sample (which comes during phase p_k), so that the total failure probability is at most $\sum_{k\geq 1} \delta_k \leq \delta$.

The quantity $\Delta(T, \eta)$ is derived from deviation bounds for bootstrap samples. Note that the projection of the hypothesis class $\mathcal{H}$ onto U_0 is bounded in size by $\mathcal{S}(\mathcal{H}, m) = O(m^d)$ [Sau72], where for a family $\mathcal{F}$ of functions $f : \mathcal{Z} \to \{0, 1\}$,

$$\mathcal{S}(\mathcal{F}, m) := \sup\{|\{(f(z_1), \ldots, f(z_m))\}| : (z_1, \ldots, z_m) \in \mathcal{Z}^m\}$$

is the mth *shatter coefficient* of $\mathcal{F}$. (This is a worst case bound; the size of the projection can be much smaller.)

Therefore we only need to bound the deviations of errors for at most $O(m^d)$ hypotheses. This observation is standard in the proof of many uniform convergence bounds [VC71], but here we use it explicitly because we are actually dealing with a fixed, finite sample U_0.

We can therefore employ large deviation inequalities to bound the deviation of $\operatorname{err}(h, T)$ from $\operatorname{err}(h, Z_0)$, where T is a random subset of Z_0.

Lemma 4.1. *Pick any $\eta \in (0, 1)$ and finite $Z_0 \subseteq \mathcal{X} \times \mathcal{Y}$. With probability at least $1 - \eta$ over the choice of a random subset T of Z_0,*

$$|\operatorname{err}(h, Z_0) - \operatorname{err}(h, T)| \leq \sqrt{\frac{\log(2M/\eta)}{2|T|}}$$

for all $h \in \mathcal{H}$, where $M := |\{(h(x) : (x, y) \in Z_0) : h \in \mathcal{H}\}|$.

Proof. An easy application of Hoeffding's inequality and the union bound. □

In light of this, we set

$$\Delta(T, \eta) := 2 \cdot \sqrt{\frac{\log(2M/\eta)}{2|T|}} \tag{4.2}$$

where M is the quantity specified in the lemma, noting that $M = O(m^d)$. The following corollary is immediate given the definition of Δ and δ_k.

Corollary 4.1. *With probability at least $1 - \delta$ over the choice of random samples $\{T_{p_k} : k \geq 1\}$,*

$$|(\operatorname{err}(h, Z_{p_k}) - \operatorname{err}(h^*, Z_{p_k})) - (\operatorname{err}(h, T_{p_k}) - \operatorname{err}(h^*, T_{p_k}))| \leq \Delta(T_{p_k}, \delta_k) \tag{4.3}$$

for all $h \in \mathcal{H}$ and all $k \geq 1$.

4.2.3 Correctness Analysis

For each p, let Z_p be the subset of labeled examples from Z_0 whose unlabeled component is in U_p. Note that $Z_0 \setminus Z_p = S_p$, where S_p (lacking the ornament) is the same as $\tilde{S}_p$, except with the true labels y swapped in for the synthesized labels $\tilde{y}$. Therefore, if $h, h' \in \mathcal{H}$ agree on how to label points in $\tilde{S}_p$, then $\operatorname{err}(h, Z_0 \setminus Z_p) = \operatorname{err}(h', Z_0 \setminus Z_p)$. This will be a key point used to prove the following lemma.

Lemma 4.2. *Assume the bound from Eq. (4.3) holds for all $h \in \mathcal{H}$ and all $k \geq 1$. For all $p \geq 0$, h^* is consistent with all examples in $\tilde{S}_p$, and*

$$\operatorname{err}(h^*, Z_p) \leq \operatorname{err}(h, Z_p) \tag{4.4}$$

for all $h \in \mathcal{H}$ consistent with examples in $\tilde{S}_p$.

Proof. By induction on p. The base case $p = 0$ is trivially true by the definitions of Z_0, $\tilde{S}_0$, and h^*. So pick $\ell \geq 0$ and assume as the inductive hypothesis that h^* is consistent with all examples in S_ℓ and that Eq. (4.4) holds for $p = \ell$. First, it is clear that $h_\ell \neq \perp$ by the inductive hypothesis, since h^* is consistent with S_ℓ. Suppose for sake of contradiction that some $(x, \tilde{y})$ is added to $\tilde{S}'$, but $h^*(x) \neq \tilde{y} = h_\ell(x)$. It must be that $h'_{x,\ell} \neq \perp$ and

$$\text{err}(h'_{x,\ell}, T_\ell) - \text{err}(h_\ell, T_\ell) > \Delta(T_\ell, \delta_k)$$

for the current value of k. Moreover, since T_ℓ is a random subset of Z_ℓ, we have by Corollary 4.1 and the definition of $h'_{x,\ell}$, that

$$\begin{aligned}\text{err}(h^*, Z_\ell) - \text{err}(h_\ell, Z_\ell) &\geq \text{err}(h^*, T_\ell) - \text{err}(h_\ell, T_\ell) - \Delta(T_\ell, \delta_k)\\ &\geq \text{err}(h'_{x,\ell}, T_\ell) - \text{err}(h_\ell, T_\ell) - \Delta(T_\ell, \delta_k)\\ &> \Delta(T_\ell, \delta_k) - \Delta(T_\ell, \delta_k) \ = 0\end{aligned}$$

so $\text{err}(h^*, Z_\ell) > \text{err}(h_\ell, Z_\ell)$, a contradiction of the inductive hypothesis. Therefore $h^*(x) = \tilde{y}$ for all $(x, \tilde{y})$ added to $\tilde{S}'$, which are those ultimately added to $\tilde{S}_{\ell+1}$. Note that each such $x \in U'$, so $x \notin U_{\ell+1}$.

Take any $h \in \mathcal{H}$ consistent with examples in $\tilde{S}_{\ell+1}$; such an h therefore agrees with h^* on $\tilde{S}_{\ell+1}$. Then $\text{err}(h, Z_0 \setminus Z_{\ell+1}) = \text{err}(h^*, Z_0 \setminus Z_{\ell+1})$, so

$$\begin{aligned}\text{err}(h, Z_0) &= \frac{|Z_0 \setminus Z_{\ell+1}|}{m} \cdot \text{err}(h, Z_0 \setminus Z_{\ell+1}) + \frac{|Z_{\ell+1}|}{m} \cdot \text{err}(h, Z_{\ell+1})\\ &= \frac{|Z_0 \setminus Z_{\ell+1}|}{m} \cdot \text{err}(h^*, Z_0 \setminus Z_{\ell+1}) + \frac{|Z_{\ell+1}|}{m} \cdot \text{err}(h, Z_{\ell+1})\\ &= \text{err}(h^*, Z_0) + \frac{|Z_{\ell+1}|}{m} \cdot (\text{err}(h, Z_{\ell+1}) - \text{err}(h^*, Z_{\ell+1})).\end{aligned}$$

Because $\text{err}(h, Z_0) \geq \text{err}(h^*, Z_0)$ by definition of h^*, it must be that $\text{err}(h, Z_{\ell+1}) \geq \text{err}(h^*, Z_{\ell+1})$. □

Theorem 4.1. *The following holds with probability at least $1 - \delta$ over the choice of random subsets generated by Reduction-based A^2. If Reduction-based A^2 returns a hypothesis h, then $\text{err}(h, Z_0) \leq \text{err}(h^*, Z_0) + \epsilon$.*

Proof. We first apply the bounds from Corollary 4.1, which hold with probability at least $1 - \delta$. Then, Lemma 4.2 implies that h^* is consistent with all examples in $\tilde{S}_p$ for all $p \geq 0$. Therefore,

$$\begin{aligned}\text{err}(h, Z_0) &= \frac{|Z_0 \setminus Z_p|}{m} \cdot \text{err}(h, Z_0 \setminus Z_p) + \frac{|Z_p|}{m} \cdot \text{err}(h, Z_p)\\ &= \frac{|Z_0 \setminus Z_p|}{m} \cdot \text{err}(h^*, Z_0 \setminus Z_p) + \frac{|Z_p|}{m} \cdot \text{err}(h, Z_p)\\ &= \text{err}(h^*, Z_0) + \frac{|Z_p|}{m} \cdot (\text{err}(h_p, Z_p) - \text{err}(h^*, Z_p))\\ &= \text{err}(h^*, Z_0) + \frac{|U_p|}{|U_0|} \cdot (\text{err}(h_p, Z_p) - \text{err}(h^*, Z_p)). \qquad (4.5)\end{aligned}$$

If a hypothesis h is returned in phase p, then either $(|U_p|/|U_0|) \cdot \Delta(T_p, \delta_k) \leq \epsilon$, or $|U_{p+1}|/|U_0| \leq \epsilon$. In the former case, we have $h = h_p = \text{LEARN}_{\mathcal{H}}(\tilde{S}_p, T_p)$. Using Eq. (4.5) and Corollary 4.1, we have

$$\begin{aligned}\operatorname{err}(h, Z_0) &= \operatorname{err}(h^*, Z_0) + \frac{|U_p|}{|U_0|} \cdot (\operatorname{err}(h_p, Z_p) - \operatorname{err}(h^*, Z_p)) \\ &\leq \operatorname{err}(h^*, Z_0) + \frac{|U_p|}{|U_0|} \cdot (\operatorname{err}(h_p, T_p) - \operatorname{err}(h^*, T_p) + \Delta(T_p, \delta_k)) \\ &\leq \operatorname{err}(h^*, Z_0) + \frac{|U_p|}{|U_0|} \cdot \Delta(T_p, \delta_k) \\ &\leq \operatorname{err}(h^*, Z_0) + \epsilon.\end{aligned}$$

In the latter case, we have $h = \text{LEARN}_{\mathcal{H}}(\tilde{S}_{p+1}, \emptyset)$, so using Eq. (4.5), we have

$$\begin{aligned}\operatorname{err}(h, Z_0) &= \operatorname{err}(h^*, Z_0) + \frac{|U_{p+1}|}{|U_0|} \cdot (\operatorname{err}(h, Z_{p+1}) - \operatorname{err}(h^*, Z_{p+1})) \\ &\leq \operatorname{err}(h^*, Z_0) + \epsilon\end{aligned}$$

as required. □

4.2.4 Discussion

Reduction-based A^2 essentially provides a particular implementation of A^2 proper using the $\text{LEARN}_{\mathcal{H}}$ primitive on a finite random sample. This is a sort of "batch mode" active learning algorithm in that all of the unlabeled data is accessed up front. One advantage of such methods is that decisions for whether to query a label can be made with full knowledge of other points that could also be labeled. Note that it is not completely clear if Reduction-based A^2, or even A^2 proper, fully exploits this advantage—we leave this as an interesting open question. On the other hand, if additional unlabeled data is acquired, it is not immediately obvious how to incorporate them into the active learning process.

In contrast to Reduction-based A^2, CAL operates by examining the data one at a time. In this "online mode" of active learning, the label of an unlabeled point is either queried upon first seeing this point, or never at all. Thus, this style of active learning is complementary in the above mentioned strength and weakness of "batch mode" active learning. In the next section, we present a more direct generalization of CAL to agnostic active learning that retains the "online" aspect. It will also improve on the label complexity guarantees in Theorem 2.3 that were afforded to A^2 proper.

4.3 An Agnostic Generalization of CAL

A more direct generalization of CAL to the agnostic setting comes from viewing CAL in reduction form (Algorithm 3.1). By replacing the PAC notions used in CAL with appropriate agnostic analogues, we arrive at the Agnostic CAL algorithm, which is specified in Algorithm 4.2. Specifically:

- In place of the reduction to finding consistent hypotheses, we use the subroutine $\text{LEARN}_{\mathcal{H}}$ to select minimize empirical error over an implicit version space.

```
Algorithm 4.2 (Agnostic CAL)
Notes: $\delta_t := \delta/(t^2+t)$ for all $t \geq 1$; see Eq. (4.10) for the definition of $\Delta$.
Initialize: $\tilde{S}_0 := \emptyset$, $T_0 := \emptyset$.
For $t = 1, 2, \ldots, n$:
    1. Obtained unlabeled data point $X_t$.
    2. Let
        (a) $h_t := \mathrm{LEARN}_{\mathcal{H}}(\tilde{S}_{t-1}, T_{t-1})$, and
        (b) $h'_t := \mathrm{LEARN}_{\mathcal{H}}(\tilde{S}_{t-1} \cup \{(X_t, -h_t(X_t))\}, T_{t-1})$.
    3. If $h'_t \neq \perp$ and
           $\mathrm{err}(h'_t, \tilde{S}_{t-1} \cup T_{t-1}) - \mathrm{err}(h_t, \tilde{S}_{t-1} \cup T_{t-1}) \leq \Delta(h'_t, h_t, \tilde{S}_{t-1} \cup T_{t-1}, \delta_{t-1})$
        (a) Then: Query $Y_t$, and set $\tilde{S}_t := \tilde{S}_{t-1}$ and $T_t := T_{t-1} \cup \{(X_t, Y_t)\}$.
        (b) Else: Set $\tilde{Y}_t := h_t(X_t)$, and $\tilde{S}_t := \tilde{S}_{t-1} \cup \{(X_t, \tilde{Y}_t)\}$ and $T_t := T_{t-1}$.
Return: $h_{n+1} := \mathrm{LEARN}_{\mathcal{H}}(\tilde{S}_n, T_n)$.
```

Figure 4.2: The Agnostic CAL algorithm.

- Instead of simply checking for the existence of consistent hypotheses, we use a robust test that compares empirical errors.

4.3.1 Deviation Bounds for Error Differences

The test used by Agnostic CAL (Step 3 in Algorithm 4.2) appears similar to the one used in Reduction-based A^2 (Step 7(b) in Algorithm 4.1). However, the test used by Reduction-based A^2 compares the empirical errors estimated from a random labeled samples, whereas this does not appear to be the case with that of Agnostic CAL: the empirical errors are computed using labels in $\tilde{S}_{t-1}$ determined by the algorithm, rather than queried (*i.e.*, drawn from the underlying distribution).

The key trick to justifying the test is to consider the deviations of *empirical error differences* from their expectations, rather than simply the deviations of empirical errors from their expectations (as was done in Lemma 4.1 for Reduction-based A^2). For any n, let

$$S_n := \{(X_i, Y_i) : (X_i, \tilde{Y}_i) \in \tilde{S}_n\}$$

be the set of labeled examples the same as $\tilde{S}_n$, except with the true labels swapped in. Note, then, that $S_n \cup T_n$ is an iid sample of n labeled examples. Consider two hypotheses h and h', both of which are consistent with a set of labeled examples $\tilde{S}_n$. Then, we have

$$\mathrm{err}(h, \tilde{S}_n \cup T_n) - \mathrm{err}(h', \tilde{S}_n \cup T_n) = \mathrm{err}(h, S_n \cup T_n) - \mathrm{err}(h', S_n \cup T_n).$$

In other words, the difference in empirical errors on $\tilde{S}_n \cup T_n$ is precisely the same as the difference in empirical errors on $S_n \cup T_n$. Therefore, we can apply bounds similar to those from Lemma 4.1, so long as we are concerned only with hypotheses that agree on $\tilde{S}_n$.

However, the bounds similar to those from Lemma 4.1 are insufficient for our purposes. This is because such bounds do not take into account variance information: specifically, the fact that deviations scale not only with the size of the random sample, but also with the variance. The variance of $\mathrm{err}(h, S_n \cup T_n)$ is $\mathrm{err}(h)(1 - \mathrm{err}(h))/n$, so the further $\mathrm{err}(h)$ is from $1/2$, the smaller the variance is. Note that if $\mathrm{err}(h) = 0$, then the deviations ought to behave like the bounds used in the analysis of CAL (*e.g.*, Eq. (3.1)). We will instead use *normalized* uniform convergence bounds [VC71], which interpolate between $O(1/\sqrt{n})$ and $O(1/n)$, depending on expectation of the random variable in question.

There is one more detail to deal with. The bound we will use deals with $\{0, 1\}$-valued functions, whereas the empirical error differences $\mathrm{err}(h, S_n \cup T_n) - \mathrm{err}(h', S_n \cup T_n)$ are the averages of $\{-1, 0, +1\}$-valued functions. We will work around this in the following manner. Let $Z_n := S_n \cup T_n$ and $\tilde{Z}_n := \tilde{S}_n \cup T_n$. Let $\mathcal{A} := \{a_{h,h'} : (h, h') \in \mathcal{H}^2\}$ and $\mathcal{B} := \{b_{h,h'} : (h, h') \in \mathcal{H}^2\}$, where

$$\begin{aligned} a_{h,h'}(x, y) &:= \mathbb{1}(h(x) \neq y \ \wedge \ h'(x) = y) \\ b_{h,h'}(x, y) &:= \mathbb{1}(h(x) = y \ \wedge \ h'(x) \neq y). \end{aligned}$$

Then, define

$$\begin{aligned} a_n(h, h') &:= \frac{1}{n} \sum_{(X_i, Y_i) \in Z_n} a_{h,h'}(X_i, Y_i) \\ b_n(h, h') &:= \frac{1}{n} \sum_{(X_i, Y_i) \in Z_n} b_{h,h'}(X_i, Y_i). \end{aligned}$$

Define $\tilde{a}_n$ and $\tilde{b}_n$ similarly, replacing Z_n with $\tilde{Z}_n$. We have

$$\begin{aligned} a_n(h, h') - b_n(h, h') &= \mathrm{err}(h, Z_n) - \mathrm{err}(h', Z_n) \\ \mathbb{E}[a_n(h, h') - b_n(h, h')] &= \mathrm{err}(h) - \mathrm{err}(h'). \end{aligned}$$

We can now state our deviation bounds for error differences.

Lemma 4.3. *Pick any $n \geq 1$ and $\eta \in (0, 1)$. Let*

$$\varepsilon_n := \frac{4}{n} \cdot \left(2d \ln \frac{2en}{d} + \ln \frac{24}{\eta} \right).$$

Let $Z_n := \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ be a set of n iid copies of (X, Y), $S_n \subseteq Z_n$ an arbitrary subset; $T_n := Z_n \setminus S_n$; $\tilde{S}_n := \{(X_i, \tilde{y}_i) : (X_i, Y_i) \in S_n\}$ for any $\tilde{y}_1, \dots, \tilde{y}_n \in \mathcal{Y}$; and $\tilde{Z}_n := \tilde{S}_n \cup T_n$. The following holds with probability at least $1 - \eta$.

For all $h \in \mathcal{H}$,

$$\begin{aligned} &-\min\left(\varepsilon_n + \sqrt{\varepsilon_n \cdot \mathrm{err}(h)},\ \sqrt{\varepsilon_n \cdot \mathrm{err}(h, Z_n)}\right) \\ &\quad \leq \mathrm{err}(h) - \mathrm{err}(h, Z_n) \leq \min\left(\sqrt{\varepsilon_n \cdot \mathrm{err}(h)},\ \varepsilon_n + \sqrt{\varepsilon_n \cdot \mathrm{err}(h, Z_n)}\right). \end{aligned} \tag{4.6}$$

For all $(h, h') \in \mathcal{H}^2$,

$$\operatorname{err}(h, Z_n) - \operatorname{err}(h', Z_n) \leq \operatorname{err}(h) - \operatorname{err}(h') + \varepsilon_n + \sqrt{\varepsilon_n a_n(h, h')} + \sqrt{\varepsilon_n b_n(h, h')}. \qquad (4.7)$$

For all $(h, h') \in \mathcal{H}^2$ *such that* h *and* h' *agree on* $\tilde{S}_n$,

$$\begin{aligned} &\operatorname{err}(h, \tilde{Z}_n) - \operatorname{err}(h', \tilde{Z}_n) \\ &\leq \operatorname{err}(h) - \operatorname{err}(h') + \varepsilon_n + \sqrt{\varepsilon_n \tilde{a}_n(h, h')} + \sqrt{\varepsilon_n \tilde{b}_n(h, h')} && (4.8) \\ &\leq \operatorname{err}(h) - \operatorname{err}(h') + \varepsilon_n + \sqrt{\varepsilon_n \operatorname{err}(h, \tilde{Z}_n)} + \sqrt{\varepsilon_n \operatorname{err}(h', \tilde{Z}_n)}. && (4.9) \end{aligned}$$

Proof. First, Lemma A.2 (with failure probability $\eta/3$) easily applies to the loss class $\{(x, y) \mapsto \mathbb{1}(h(x) \neq y) : h \in \mathcal{H}\}$ to give Eq. (4.6). Next, note that $\max(\mathcal{S}(\mathcal{A}, m), \mathcal{S}(\mathcal{B}, m)) \leq \mathcal{S}(\mathcal{H}, m)^2 \leq (em/d)^d$ by Sauer's Lemma [Sau72]. Therefore, we can apply Lemma A.2 to the class $\mathcal{A}$ and $\mathcal{B}$ (with failure probability $\eta/3$ each) to give Eq. (4.7). To get Eq. (4.8) from Eq. (4.7), we just notice that

$$\begin{aligned} \operatorname{err}(h, \tilde{Z}_n) - \operatorname{err}(h', \tilde{Z}_n) &= \operatorname{err}(h, Z_n) - \operatorname{err}(h', Z_n) \\ \tilde{a}_n(h, h') &= a_n(h, h') \\ \tilde{b}_n(h, h') &= b_n(h, h') \end{aligned}$$

for all $(h, h') \in \mathcal{H}^2$ that agree on $\tilde{S}_n$. Finally, to get Eq. (4.9), we use the facts $\tilde{a}_n(h, h') \leq \operatorname{err}(h, \tilde{Z}_n)$ and $\tilde{b}_n(h, h') \leq \operatorname{err}(h', \tilde{Z}_n)$. □

In light of Lemma 4.3, we will define Δ as

$$\Delta(h, h', \tilde{Z}, \eta) := \varepsilon_{|\tilde{Z}|} + \sqrt{\varepsilon_{|\tilde{Z}|} \operatorname{err}(h, \tilde{Z})} + \sqrt{\varepsilon_{|\tilde{Z}|} \operatorname{err}(h', \tilde{Z})} \qquad (4.10)$$

where

$$\varepsilon_n := \frac{4}{n} \cdot \left(2d \ln \frac{2en}{d} + \ln \frac{24}{\eta} \right). \qquad (4.11)$$

We use $\delta_t := \delta/(t^2 + t)$ so that $\sum_{t \geq 1} \delta_t \leq \delta$.

We remark that our choice of the threshold function Δ is based on the agnostic learning model. In general, however, it can be based on any computable deviation bound suitable for the learning model.

4.3.2 Correctness Analysis

The following is an agnostic analogue of Theorem 2.1—the consistency guarantee for CAL in the PAC setting. Here, we prove that the version space implicitly defined by the set $\tilde{S}_t$ always contains the optimal hypothesis h^*.

Lemma 4.4. *Assume the bound from Eq.* (4.9) *holds for all* $(h, h') \in \mathcal{H}^2$ *and all* $n \geq 1$, *using* $\eta = \delta_n$ *when applied to* Z_n. *The hypothesis* h^* *is consistent with all examples in* $\tilde{S}_n$ *for all* $n \geq 0$.

Proof. First, note that the bounds from Eq. (4.9) trivially hold for $n = 0$, so we have by assumption that they hold for all $n \geq 0$. Now we proceed by induction on n. The base case of $n = 0$ holds trivially since $\tilde{S}_0 = \emptyset$. So pick any $n \geq 1$ and assume as the inductive hypothesis that h^* is consistent with $\tilde{S}_{n-1}$. Suppose, in iteration n, that $(X_n, \tilde{Y}_n)$ is added to $\tilde{S}_n$. If $h'_n = \perp$, then every $h \in \mathcal{H}$ consistent with $\tilde{S}_{n-1}$ must label X_n the same as $h_n(X_n)$. By the inductive hypothesis, it must be that $h^*(X_n) = h_n(X_n) = \tilde{Y}_n$. If $h'_n \neq \perp$, then

$$\operatorname{err}(h'_n, \tilde{Z}_{n-1}) - \operatorname{err}(h_n, \tilde{Z}_{n-1}) > \varepsilon_{n-1} + \sqrt{\varepsilon_{n-1} \cdot \operatorname{err}(h'_n, \tilde{Z}_{n-1})} + \sqrt{\varepsilon_{n-1} \cdot \operatorname{err}(h_n, \tilde{Z}_{n-1})}.$$

In particular, $\operatorname{err}(h'_n, \tilde{Z}_{n-1}) \geq \varepsilon_{n-1}$. Suppose, for sake of contradiction, that $h^*(X_n) \neq h_n(X_n)$. Then $\operatorname{err}(h^*, \tilde{Z}_n) \geq \operatorname{err}(h'_n, \tilde{Z}_n) > \varepsilon_{n-1}$ by definition of h'_n (by the inductive hypothesis, h^* is consistent with $\tilde{S}_n$, yet $\text{LEARN}_{\mathcal{H}}$ returns h'_n in preference to it). Therefore,

$$\begin{aligned}
&\operatorname{err}(h^*, \tilde{Z}_{n-1}) - \operatorname{err}(h_n, \tilde{Z}_{n-1}) \\
&= \operatorname{err}(h^*, \tilde{Z}_{n-1}) - \operatorname{err}(h'_n, \tilde{Z}_{n-1}) + \operatorname{err}(h'_n, \tilde{Z}_{n-1}) - \operatorname{err}(h_n, \tilde{Z}_{n-1}) \\
&> \sqrt{\operatorname{err}(h'_n, \tilde{Z}_{n-1})} \left(\sqrt{\operatorname{err}(h^*, \tilde{Z}_{n-1})} - \sqrt{\operatorname{err}(h'_n, \tilde{Z}_{n-1})} \right) \\
&\quad + \varepsilon_{n-1} + \sqrt{\varepsilon_{n-1} \cdot \operatorname{err}(h'_n, \tilde{Z}_{n-1})} + \sqrt{\varepsilon_{n-1} \cdot \operatorname{err}(h_n, \tilde{Z}_{n-1})} \\
&> \sqrt{\varepsilon_{n-1} \cdot \operatorname{err}(h^*, \tilde{Z}_{n-1})} - \sqrt{\varepsilon_{n-1} \cdot \operatorname{err}(h'_n, \tilde{Z}_{n-1})} \\
&\quad + \varepsilon_{n-1} + \sqrt{\varepsilon_{n-1} \cdot \operatorname{err}(h'_n, \tilde{Z}_{n-1})} + \sqrt{\varepsilon_{n-1} \cdot \operatorname{err}(h_n, \tilde{Z}_{n-1})} \\
&= \varepsilon_{n-1} + \sqrt{\varepsilon_{n-1} \cdot \operatorname{err}(h^*, \tilde{Z}_{n-1})} + \sqrt{\varepsilon_{n-1} \cdot \operatorname{err}(h_n, \tilde{Z}_{n-1})}.
\end{aligned}$$

Now, the bounds from Eq. (4.9) implies $\operatorname{err}(h^*) > \operatorname{err}(h_n)$, a contradiction. □

Lemma 4.4, together with the deviation bounds in Lemma 4.3, immediately implies that Agnostic CAL has essentially the same sample complexity bound as a supervised learner based on empirical risk minimization.

Theorem 4.2. *With probability at least* $1 - \delta$,

$$\operatorname{err}(h_{n+1}) \leq \operatorname{err}(h^*) + O\left(\sqrt{\operatorname{err}(h^*) \cdot \frac{d \log n + \log(1/\delta)}{n}} + \frac{d \log n + \log(1/\delta)}{n} \right).$$

Proof. Follows from the deviation bounds in Lemma 4.3, the consistency guarantee from Lemma 4.4, and some simple algebraic manipulations. □

The error bound in Theorem 4.2 differs from the error bound of a fully-supervised learner (see Eq. (4.1)) by constant factors.

4.3.3 Label Complexity Analysis

We now give a bound on the number of labels requested by Agnostic CAL after n iterations. This will recover the label complexity analysis for Reduction-based CAL (Theorem 3.1) when $\operatorname{err}(h^*) = 0$.

Lemma 4.5. *Assume the bounds from Eq.* (4.6) *and Eq.* (4.9) *hold for all* $(h, h') \in \mathcal{H}^2$ *and all* $n \geq 1$*, using* $\eta = \delta_n$ *when applied to* Z_n*. There exists a universal constant* $C \in (0, 25)$ *such that the following holds. Pick any* $n \geq 1$*, and let* $Q_{n+1} \in \{0, 1\}$ *be the random variable that indicates if* Y_{n+1} *is queried. Then*

$$\mathbb{E}[Q_{n+1}] \leq \theta \cdot \left((2 + \lambda) \cdot \mathrm{err}(h^*) + C \cdot \left(1 + \frac{1}{\lambda}\right) \cdot \frac{4}{n} \cdot \left(2d \ln \frac{2en}{d} + 2 \ln \frac{24n}{\delta}\right)\right)$$

for any $\lambda > 0$.

Proof. We apply Lemma 4.4 to ensure that h^* is consistent with $\tilde{S}_n$. Agnostic CAL queries Y_{n+1} iff

$$\mathrm{err}(h'_{n+1}, \tilde{Z}_n) - \mathrm{err}(h_{n+1}, \tilde{Z}_n) \leq \varepsilon_n + \sqrt{\varepsilon_n \cdot \mathrm{err}(h'_{n+1}, \tilde{Z}_n)} + \sqrt{\varepsilon_n \cdot \mathrm{err}(h_{n+1}, \tilde{Z}_n)}$$

where ε_n is defined in Eq. (4.11). Assume that $h^*(X_{n+1}) \neq h'_{n+1}(X_{n+1})$—this is without loss of generality since, as we could otherwise exchange the roles of h_{n+1} and h'_{n+1} in the subsequent argument. The left-hand side of the above inequality is bounded below using

$$\begin{aligned} \mathrm{err}(h'_{n+1}, \tilde{Z}_n) - \mathrm{err}(h_{n+1}, \tilde{Z}_n) &\geq \mathrm{err}(h'_{n+1}, \tilde{Z}_n) - \mathrm{err}(h^*, \tilde{Z}_n) \\ &= \mathrm{err}(h'_{n+1}, Z_n) - \mathrm{err}(h^*, Z_n), \end{aligned}$$

and the right-hand side is bounded above using

$$\begin{aligned} &\varepsilon_n + \sqrt{\varepsilon_n \cdot \mathrm{err}(h'_{n+1}, \tilde{Z}_n)} + \sqrt{\varepsilon_n \cdot \mathrm{err}(h_{n+1}, \tilde{Z}_n)} \\ &\leq \varepsilon_n + \sqrt{\varepsilon_n \cdot \mathrm{err}(h'_{n+1}, \tilde{Z}_n)} + \sqrt{\varepsilon_n \cdot \mathrm{err}(h^*, \tilde{Z}_n)} \\ &\leq \varepsilon_n + \sqrt{\varepsilon_n \cdot \mathrm{err}(h'_{n+1}, Z_n)} + \sqrt{\varepsilon_n \cdot \mathrm{err}(h^*, Z_n)}; \end{aligned}$$

the last inequality follows because $\mathrm{err}(h, \tilde{Z}_n) \leq \mathrm{err}(h, Z_n)$ for all h is consistent with $\tilde{S}_n$. Therefore

$$\mathrm{err}(h'_{n+1}, Z_n) - \mathrm{err}(h^*, Z_n) \leq \varepsilon_n + \sqrt{\varepsilon_n \cdot \mathrm{err}(h'_{n+1}, Z_n)} + \sqrt{\varepsilon_n \cdot \mathrm{err}(h^*, Z_n)}.$$

Now, we use the bounds from Eq. (4.6) to give

$$\begin{aligned} \mathrm{err}(h'_{n+1}) - \mathrm{err}(h^*) \leq \mathrm{err}(h'_{n+1}, Z_n) &- \mathrm{err}(h^*, Z_n) \\ &+ \varepsilon_n + \sqrt{\varepsilon_n \cdot \mathrm{err}(h'_{n+1}, Z_n)} + \sqrt{\varepsilon_n \cdot \mathrm{err}(h^*, Z_n)}. \end{aligned}$$

Combining the previous two inequalities, applying Eq. (4.6) to the empirical error terms $\mathrm{err}(h'_{n+1}, Z_n)$ and $\mathrm{err}(h^*, Z_n)$ inside the square-roots, and simplifying the quadratic inequalities gives

$$\mathrm{err}(h'_{n+1}) \leq (1 + \lambda) \cdot \mathrm{err}(h^*) + C \cdot \left(1 + \frac{1}{\lambda}\right) \cdot \varepsilon_n$$

for any $\lambda > 0$. Here $C \in (0, 25)$ is some universal constant (which is almost certainly loose). By the triangle inequality, we have

$$\begin{aligned}\rho(h^*, h_{n+1}) &\leq \mathrm{err}(h^*) + \mathrm{err}(h'_{n+1}) \\ &\leq (2+\lambda)\cdot \mathrm{err}(h^*) + C \cdot \left(1 + \frac{1}{\lambda}\right) \cdot \varepsilon_n =: r_n.\end{aligned}$$

Therefore, since $h^*(X_{n+1}) = h_{n+1}(X_{n+1}) \neq h'_{n+1}(X_{n+1})$ (by assumption), it must be that $X_{n+1} \in \mathcal{R}(h^*, r_n)$. The result now follows from the definition of the disagreement coefficient θ:

$$\mathbb{E}[Q_{n+1}] = \mathbb{E}[\mathbb{E}[Q_{n+1}|Z_n, X_{n+1}]] \leq \mathbb{E}[\Pr(X_{n+1} \in \mathcal{R}(h^*, r_n))] \leq \theta \cdot r_n.$$

□

Theorem 4.3. *There exists a universal constant $C \in (0, 25)$ such that the following holds. Conditioned on an event that occurs with probability at least $1 - \delta$, the expected number of labels queried by Agnostic CAL after n iterations is at most*

$$\begin{aligned}&1 + 2\cdot\theta\cdot\mathrm{err}(h^*)\cdot(n-1) \\ &\qquad + 4\cdot\theta\cdot\sqrt{C\cdot\mathrm{err}(h^*)\cdot\left(2d\ln\frac{2en}{d} + 2\ln\frac{24n}{\delta}\right)\cdot(n-1)\cdot\ln n} \\ &\qquad\qquad + 4C\cdot\theta\cdot\left(2d\ln\frac{2en}{d} + 2\ln\frac{24n}{\delta}\right)\cdot\ln n.\end{aligned}$$

Proof. Assume Y_1 is always queried. Apply Lemmas 4.3 and 4.5, and linearity of expectation to bound $\mathbb{E}[Q_2 + \ldots + Q_n]$. Then optimize over the choice of λ. □

The bound here implies a label complexity that is essentially a factor of θ smaller than the label complexity guarantee for A^2 (Theorem 2.3).

4.3.4 Discussion

Relation to Previous and Subsequent Work

In many ways, Agnostic CAL can be seen as a direct generalization of Reduction-based CAL that imports the robustness of A^2 using deviation bounds. Note that if it is assumed that $\mathrm{err}(h^*) = 0$, then it would be enough to set $\Delta \approx O(d/n)$ in iteration n, and we would still recover the same correctness and label complexity guarantees as those afforded to CAL. Thus, the setting of Δ in Eq. (4.10) is specifically designed to handle the adversarial noise that is allowed in the agnostic learning model. The influence of A^2 and Reduction-based A^2 is clear, as Δ is based on a deviation bound for error differences.

In subsequent work, Hanneke has shown that by using a more sophisticated setting of Δ based on Rademacher complexities, Agnostic CAL can yield improved label complexity bounds under a different noise model [Han09]. A very similar algorithm is analyzed using localized Rademacher complexities by Koltchinskii for similar label complexity improvements [Kol09]. We leave as an open question as to whether the setting of Δ prescribed in

Eq. (4.10) can be shown to adapt to these noise models. It would also be interesting to develop variants of Agnostic CAL that operate in even more adversarial settings, such as the malicious noise model or mistake-bound models. Finally, a similar algorithm that also used importance-weights was developed by Beygelzimer *et al* for certain loss functions such as logistic loss [BDL09].

Version Spaces

Thus far, the algorithms described (including the above mentioned subsequent work) are all based on either an explicit or implicit version space. That is, the hypothesis returned by the algorithm is selected from a restricted subset of the hypothesis class. In Reduction-based A^2 and Agnostic CAL, we have encapsulated enforcement of the hard constraints defining the implicit version space in the $\text{LEARN}_{\mathcal{H}}$ subroutine. However, hard constraints can make the algorithm brittle, as a single mishap by the algorithm can potentially evict the optimal hypothesis h^*. Moreover, hard constraints can be computationally cumbersome to enforce, especially for complex hypothesis classes. Instead, we would prefer algorithms that avoid explicit enforcement of a version space when selecting a hypothesis to return. This will be explored in the next chapter.

4.4 Bibliographic Notes

This chapter is based on joint work with Sanjoy Dasgupta and Claire Monteleoni titled “A General Agnostic Active Learning Algorithm”, published in the proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems in 2007 [DHM07]. The dissertation author was the primary investigator and author of this paper.

Chapter 5

Reduction to Agnostic Learning II: Error Minimization Oracles

We describe agnostic active learning algorithms that are not explicitly based on the version space approach. These algorithms use error minimization oracles that are simpler than the LEARN$_{\mathcal{H}}$ subroutine of A^2 and Agnostic CAL.

5.1 Introduction

The A^2 and Agnositc CAL algorithms use a set of labeled examples $\tilde{S}$ together with hard constraints (enforced by the subroutine LEARN$_{\mathcal{H}}$) in order to maintain an implicit version space, and hypotheses selected by these algorithms are always chosen from this space. The approach ensures a simple monotonicity property $\mathcal{H} = \mathcal{V}_0 \supseteq \mathcal{V}_1 \supseteq \mathcal{V}_2 \supseteq \ldots$ that appears to both simplify and sharpen the label complexity analysis: the deviation bounds for error differences only need to apply to pairs of hypotheses within the version space, and the restriction to a subset of hypothesis class yields a tighter bound on the size of the disagreement region.

Strict adherence to an implicit version space, however, has potential drawbacks. The first is the computational difficulty of respecting the hard constraints that define the version space. For instance, with the class of linear separators, the version space is the intersection of several half-spaces, one per example in $\tilde{S}$. Although some of the constraints will likely be redundant and thus safe to ignored, the constraints nonetheless complicate the implementation of the LEARN$_{\mathcal{H}}$ subroutine. Moreover, the difficulty is only increased with more complicated predictors (*e.g.*, decision trees, neural networks). The plausibility of LEARN$_{\mathcal{H}}$ matching our abstraction of practical supervised learning algorithm is therefore jeopardized by this computational difficulty.

The second drawback of the implicit version space is the danger of evicting the optimal hypothesis h^*. The same monotonicity property that appeared to be a blessing for A^2 and Agnostic CAL is also a liability in this sense. The analysis of A^2 and Agnostic CAL proves that h^* is never evicted, but only with the specified choice of the threshold function Δ. In practice, the large constants in the definition of Δ may render nil the potential benefits of active learning, but a more optimistic choice may be dangerous when coupled with a strict

```
Algorithm 5.1 (Oracular CAL)
Notes: $\delta_t := \delta/(t^2 + t)$ for all $t \geq 1$; see Eq. (5.3) for the definition of $\Delta$.
Initialize: $\tilde{S}_0 := \emptyset$, $T_0 := \emptyset$.
For $t = 1, 2, \ldots, n$:
    1. Obtained unlabeled data point $X_t$.
    2. Let
        (a) $h_t := \mathrm{LEARN}_{\mathcal{H}}(\emptyset, \tilde{S}_{t-1} \cup T_{t-1})$, and
        (b) $h'_t := \mathrm{LEARN}_{\mathcal{H}}(\{(X_t, -h_t(X_t))\}, \tilde{S}_{t-1} \cup T_{t-1})$.
    3. If $h'_t \neq \perp$ and
           $\mathrm{err}(h'_t, \tilde{S}_{t-1} \cup T_{t-1}) - \mathrm{err}(h_t, \tilde{S}_{t-1} \cup T_{t-1}) \leq \Delta(h_t, \tilde{S}_{t-1}, T_{t-1}, \delta_{t-1})$
        (a) Then: Query $Y_t$, and set $\tilde{S}_t := S_{t-1}$ and $T_t := T_{t-1} \cup \{(X_t, Y_t)\}$.
        (b) Else: Set $\tilde{Y}_t := h_t(X_t)$, and $\tilde{S}_t := \tilde{S}_{t-1} \cup \{(X_t, \tilde{Y}_t)\}$ and $T_t := T_{t-1}$.
Return: $h_{n+1} := \mathrm{LEARN}_{\mathcal{H}}(\tilde{Z}_n)$.
```

Figure 5.1: The Oracular CAL algorithm.

version space.

We address both of these drawbacks in this chapter by developing algorithms that (i) rely on an error minimization oracle simpler than $\mathrm{LEARN}_{\mathcal{H}}$, and (ii) avoid strict adherence to a version space.

5.2 A Modification of Agnostic CAL

Our first algorithm (Algorithm 5.1) is a simple modification of Agnostic CAL; we call this new algorithm Oracular CAL. Relative to Agnostic CAL, the primary differences are:

1. The threshold function Δ treats the sets $\tilde{S}_t$ and T_t separately (rather than together as $\tilde{S}_t \cup T_t$, as in Agnostic CAL).

2. The use of the $\mathrm{LEARN}_{\mathcal{H}}$ subroutine is restricted in that at most one hard constraint is used in each invocation.

   Moreover, the final hypothesis returned is obtained via a call to $\mathrm{LEARN}_{\mathcal{H}}$ with no hard constraints.

The restricted use of $\mathrm{LEARN}_{\mathcal{H}}$ makes it a more plausible abstraction of standard supervised learning algorithms. In fact, all of the hypotheses h_t—and therefore the hypothesis finally returned by the algorithm—are obtained using standard empirical risk minimization without any hard constraints; only the "alternative" hypotheses h'_t are obtained using a single hard constraint. In this sense, the version space approach is almost entirely avoided by Oracular CAL.

However, this relaxation appears to come at a cost in terms of the formal label complexity analysis. The cause of this is related to the specification of the threshold function Δ. The deviation bound (Lemma 4.3) that underlies the choice of Δ for Agnostic CAL is only valid for error differences between hypotheses that agree on the set of examples $\tilde{S}_t$. However, the hypotheses selected by Oracular CAL are not subject to hard constraints on the set $\tilde{S}_t$, so the same deviation bound cannot be applied to such hypotheses. Instead, Oracular CAL (and its analysis) will rely on a different, conservative bound that appears to lead to a somewhat worse label complexity guarantee.

To simplify the exposition, we will assume that for each $x \in \mathcal{X}$, there exists some $h, h' \in \mathcal{H}$ such that $h(x) = +1$ and $h'(x) = -1$. In other words, the entire hypothesis class $\mathcal{H}$ does not completely agree on any single data point. This is without loss of generality, because data points for which there is no disagreement in the entirety of $\mathcal{H}$ have no contribution to the error relative to the optimal hypothesis $h^* \in \mathcal{H}$.

5.2.1 A Conservative Threshold

Similar to Agnostic CAL, we will base our threshold function Δ on a deviation bound. Here, we will derive a bound that is normalized by the disagreement with the optimal hypothesis h^*.

First, recall the following definitions from the analysis of Agnostic CAL.

- S_n: the set $\tilde{S}_n$ with the true labels swapped in.
- $Z_n := S_n \cup T_n$ and $\tilde{Z}_n := \tilde{S}_n \cup T_n$.
- $a_{h,h'}(x, y) := \mathbb{1}(h(x) \neq y \ \wedge \ h'(x) = y)$.
- $b_{h,h'}(x, y) := \mathbb{1}(h(x) = y \ \wedge \ h'(x) \neq y)$.
- $a_n(h, h') := (1/n) \sum_{(X_i,Y_i) \in Z_n} a_{h,h'}(X_i, Y_i)$.
- $b_n(h, h') := (1/n) \sum_{(X_i,Y_i) \in Z_n} b_{h,h'}(X_i, Y_i)$.

Note that

$$\begin{aligned} a_n(h, h') - b_n(h, h') &= \mathrm{err}(h, Z_n) - \mathrm{err}(h', Z_n) \\ \mathbb{E}[a_n(h, h') - b_n(h, h')] &= \mathrm{err}(h) - \mathrm{err}(h') \\ a_n(h, h') + b_n(h, h') &= \rho_n(h, h') \\ \mathbb{E}[a_n(h, h') + b_n(h, h')] &= \rho(h, h') \end{aligned}$$

where

$$\rho_n(h, h') := \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}(h(X_i) \neq h'(X_i))$$

is the empirical disagreement between h and h'.

Lemma 5.1. *Pick any $n \geq 1$ and $\eta \in (0,1)$. Let*

$$\varepsilon_n := \frac{4}{n} \cdot \left(d \ln \frac{2en}{d} + \ln \frac{16}{\eta} \right).$$

Let $Z_n := \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ be a set of n iid copies of (X, Y). Fix any $h^* \in \mathcal{H}$. The following holds with probability at least $1 - \eta$. For all $h \in \mathcal{H}$,

$$\begin{aligned} &|(\mathrm{err}(h, Z_n) - \mathrm{err}(h^*, Z_n)) - (\mathrm{err}(h) - \mathrm{err}(h^*))| \\ &\qquad \leq \varepsilon_n + \min\left(\sqrt{2\varepsilon_n \rho_n(h, h^*)},\ \sqrt{2\varepsilon_n \rho(h, h^*)} \right) \end{aligned} \tag{5.1}$$

and

$$|\rho_n(h, h^*) - \rho(h, h^*)| \leq \varepsilon_n + \min\left(\sqrt{2\varepsilon_n \rho_n(h, h^*)},\ \sqrt{2\varepsilon_n \rho(h, h^*)} \right). \tag{5.2}$$

Proof. We apply Lemma A.2 to classes $\{a_{h,h^*} : h \in \mathcal{H}\}$ and $\{b_{h,h^*} : h \in \mathcal{H}\}$ (with failure probability $\eta/2$ each), and then use the fact that $\sqrt{x} + \sqrt{y} \leq \sqrt{2(x+y)}$ for nonnegative x and y. □

It may not be obvious why the above deviation bounds are useful for deriving a threshold function Δ, since the bound quantities are not obviously computable by the algorithm. However, recall that in the correctness analysis Agnostic CAL, we were able to prove that $h^*(X_i) = \tilde{Y}_i$ whenever the algorithm avoided querying the label Y_i. It turns out that we will be able to prove the same guarantee for Oracular CAL; so assume for now that it is possible to compute $\mathbb{1}(h(X_i) \neq h^*(X_i))$ for all $(X_i, \tilde{Y}_i) \in \tilde{S}_n$. Then, the only trouble that remains is computing $\mathbb{1}(h(X_i) \neq h^*(X_i))$ for $(X_i, Y_i) \in T_n$. For these examples, we can use a pessimistic bound $\mathbb{1}(h(X_i) \neq h^*(X_i)) \leq 1$ that assumes disagreement with h^*. Therefore, we have the following computable bound on $\rho_n(h, h^*)$:

$$\begin{aligned} \rho_n(h, h^*) &\leq \frac{1}{n} \cdot \left(|T_n| + \sum_{(X_i, \tilde{Y}_i) \in \tilde{S}_n} \mathbb{1}(h(X_i) \neq h^*(X_i)) \right) \\ &= \frac{1}{n} \cdot \left(|T_n| + \sum_{(X_i, \tilde{Y}_i) \in \tilde{S}_n} \mathbb{1}(h(X_i) \neq \tilde{Y}_i) \right) \end{aligned}$$

where the equality uses the assumption that $\tilde{Y}_i = h^*(X_i)$. This suggests that a suitable setting of Δ is

$$\Delta(h, \tilde{S}, T, \eta) := \varepsilon_{|\tilde{S} \cup T|} + \sqrt{2\varepsilon_{|\tilde{S} \cup T|} \cdot \frac{1}{|\tilde{S} \cup T|} \cdot \left(|T| + \sum_{(x, \tilde{y}) \in \tilde{S}} \mathbb{1}(h(x) \neq \tilde{y}) \right)} \tag{5.3}$$

where

$$\varepsilon_n := \frac{4}{n} \cdot \left(d \ln \frac{2en}{d} + \ln \frac{16}{\eta} \right). \tag{5.4}$$

We use $\delta_t := \delta/(t^2 + t)$ so that $\sum_{t \geq 1} \delta_t \leq \delta$.

5.2.2 Correctness Analysis

The analysis of Oracular CAL begins with a lemma that captures the following intuition. If some of the labels in a data set are replaced by the labels assigned by a hypothesis h^*, then h^* only appears more attractive compared to other hypothesis. This is because the substitutions penalizes hypotheses that disagree with h^* on the examples where the change is made.

Lemma 5.2. *Pick any $h^* : \mathcal{X} \to \mathcal{Y}$ and any $S, T \subseteq \mathcal{X} \times \mathcal{Y}$. If $\tilde{S} := \{(x, h^*(x)) : (x, y) \in S\}$, then*

$$\operatorname{err}(h, \tilde{S} \cup T) - \operatorname{err}(h^*, \tilde{S} \cup T) \ \geq \ \operatorname{err}(h, S \cup T) - \operatorname{err}(h^*, S \cup T) \tag{5.5}$$

for all $h : \mathcal{X} \to \mathcal{Y}$.

Proof. It suffices to show that

$$\mathbb{1}(h(x) \neq \tilde{y}) - \mathbb{1}(h^*(x) \neq \tilde{y}) \ \geq \ \mathbb{1}(h(x) \neq y) - \mathbb{1}(h^*(x) \neq y)$$

whenever $(x, y) \in S$ and $h(x) = y \neq \tilde{y} = h^*(x)$. Note that

$$-\mathbb{1}(h^*(x) \neq \tilde{y}) \ = \ 0 \ \geq \ -1 \ = \ -\mathbb{1}(h^*(x) \neq y).$$

Since $h(x) = y$, we have

$$\mathbb{1}(h(x) \neq \tilde{y}) \ = \ 1 \ \geq \ 0 \ = \ \mathbb{1}(h(x) \neq y).$$

Combining these inequalities completes the proof. □

Next, we prove an analogue of Lemma 4.4, which states that the optimal hypothesis h^* agrees with the synthesized labels in $\tilde{S}_n$.

Lemma 5.3. *Assume the bound from Eq. (5.1) holds for all $h \in \mathcal{H}$ and all $n \geq 1$, using $\eta = \delta_n$ when applied to Z_n. The hypothesis h^* is consistent with all examples in $\tilde{S}_n$ for all $n \geq 0$.*

Proof. First, note that the bounds from Eq. (5.1) trivially hold for $n = 0$, so we have by assumption that they hold for all $n \geq 0$. Now we proceed by induction on n. The base case of $n = 0$ holds trivially since $\tilde{S}_0 = \emptyset$. So pick any $n \geq 1$ and assume as the inductive hypothesis that h^* is consistent with $\tilde{S}_{n-1}$. A consequence of this is that the deviation of $\operatorname{err}(h_n, S_{n-1} \cup T_{n-1}) - \operatorname{err}(h^*, S_{n-1} \cup T_{n-1})$ below its mean is bounded by $\Delta(h_n, \tilde{S}_{n-1}, T_{n-1}, \delta_{n-1})$. That is,

$$\begin{aligned}&(\operatorname{err}(h_n) - \operatorname{err}(h^*)) - (\operatorname{err}(h_n, Z_{n-1}) - \operatorname{err}(h^*, Z_{n-1})) \\ &\quad \leq \ \varepsilon_n + \sqrt{2\varepsilon_n \rho_n(h, h^*)} \ \leq \ \Delta(h_n, \tilde{S}_{n-1}, T_{n-1}, \delta_{n-1})\end{aligned} \tag{5.6}$$

where the first inequality follows from the deviation bound in Eq. (5.1); and the second follows from the inductive hypothesis $\tilde{Y}_i = h^*(X_i)$ for all $(X_i, \tilde{Y}_i) \in \tilde{S}_{n-1}$, together with the conservative bound $\mathbb{1}(h_n(X_i) \neq h^*(X_i)) \leq 1$ for all $(X_i, Y_i) \in T_{n-1}$.

Suppose the label Y_n is not queried, so $(X_n, \tilde{Y}_n) \in \tilde{S}_n$. In this case,

$$\operatorname{err}(h_n', \tilde{Z}_{n-1}) - \operatorname{err}(h_n, \tilde{Z}_{n-1}) > \Delta(h_n, \tilde{S}_{n-1}, T_{n-1}, \delta_n). \tag{5.7}$$

It suffices to show that $\mathrm{err}(h'_n, \tilde{Z}_{n-1}) > \mathrm{err}(h^*, \tilde{Z}_{n-1})$; this implies $\tilde{Y}_n = h_n(x_n) = h^*(x_n)$ because h'_n minimizes $\mathrm{err}(h'_n, \tilde{Z}_{n-1})$ over all $h \in \mathcal{H}$ with $h(X_n) \neq h_n(X_n)$. Indeed,

$$\begin{aligned}
&\mathrm{err}(h'_n, \tilde{Z}_{n-1}) - \mathrm{err}(h^*, \tilde{Z}_{n-1}) \\
&= \mathrm{err}(h'_n, \tilde{Z}_{n-1}) - \mathrm{err}(h_n, \tilde{Z}_{n-1}) + \mathrm{err}(h_n, \tilde{Z}_{n-1}) - \mathrm{err}(h^*, \tilde{Z}_{n-1}) \\
&\geq \mathrm{err}(h'_n, \tilde{Z}_{n-1}) - \mathrm{err}(h_n, \tilde{Z}_{n-1}) + \mathrm{err}(h_n, Z_{n-1}) - \mathrm{err}(h^*, Z_{n-1}) \\
&> \Delta(h_n, \tilde{S}_{n-1}, T_{n-1}, \delta_n) + \mathrm{err}(h_n, Z_{n-1}) - \mathrm{err}(h^*, Z_{n-1}) \\
&\geq \mathrm{err}(h_n) - \mathrm{err}(h^*) \\
&\geq 0.
\end{aligned}$$

Above, the inequalities follow (respectively) from Lemma 5.2 (with the inductive hypothesis), Eq. (5.7), Eq. (5.6), and the definition of h^*. □

Theorem 5.1. *With probability at least* $1 - \delta$*,*

$$\mathrm{err}(h_{n+1}) \leq \mathrm{err}(h^*) + O\left(\frac{d \log n + \log(1/\delta)}{n} + \sqrt{\mathrm{err}(h^*) \cdot \frac{d \log n + \log(1/\delta)}{n}}\right).$$

Proof. We apply the bounds from Lemma 5.1 for all $n \geq 1$, using $\eta = \delta_n$ when applied to Z_n. These bounds hold with probability at least $1-\delta$; we henceforth condition on this event. By Lemma 5.2 and Lemma 5.3, we have

$$\mathrm{err}(h_{n+1}, Z_n) - \mathrm{err}(h^*, Z_n) \;\leq\; \mathrm{err}(h_{n+1}, \tilde{Z}_n) - \mathrm{err}(h^*, \tilde{Z}_n) \;\leq\; 0.$$

Now, using the deviation bounds,

$$\begin{aligned}
\mathrm{err}(h_{n+1}) - \mathrm{err}(h^*) &\leq \varepsilon_n + \sqrt{2\varepsilon_n \rho(h_{n+1}, h^*)} \\
&\leq \varepsilon_n + \sqrt{2\varepsilon_n(\mathrm{err}(h_{n+1}) + \mathrm{err}(h^*))} \\
&\leq \varepsilon_n + \sqrt{2\varepsilon_n \,\mathrm{err}(h_{n+1})} + \sqrt{2\varepsilon_n \,\mathrm{err}(h^*)}
\end{aligned}$$

where the second inequality follows from the triangle inequality. Solving the quadratic inequality for $\mathrm{err}(h_{n+1})$ implies

$$\begin{aligned}
\mathrm{err}(h_{n+1}) - \mathrm{err}(h^*) &\leq 2\varepsilon_n + \sqrt{2\varepsilon_n(\mathrm{err}(h^*) + \sqrt{2\varepsilon_n \,\mathrm{err}(h^*)})} \\
&\leq 2\varepsilon_n + \sqrt{2\varepsilon_n \,\mathrm{err}(h^*)} + \sqrt{2\varepsilon_n \sqrt{2\varepsilon_n \,\mathrm{err}(h^*)}} \\
&\leq (2 + 1/\sqrt{2})\varepsilon_n + (3/2)\sqrt{2\varepsilon_n \,\mathrm{err}(h^*)}
\end{aligned}$$

where we have used the fact $2\sqrt{xy} \leq x + y$ in the last step. □

Once again, we have a basic consistency guarantee implies a label complexity bound for Oracular CAL no worse (up to constants) than that of a fully-supervised learner (see Eq. (4.1)).

5.2.3 Label Complexity Analysis

We now bound the number of labels requested by Oracular CAL after n iterations. First, we show a bound on the threshold value $\Delta(h_{n+1}, \tilde{S}_n, T_n, \delta_n)$.

Lemma 5.4. *Assume the bounds from Eq.* (5.1) *and Eq.* (5.2) *hold for all $h \in \mathcal{H}$ and all $n \geq 1$, using $\eta = \delta_n$ when applied to Z_n. For all $n \geq 1$,*

$$\Delta(h_{n+1}, \tilde{S}_n, T_n, \delta_n) \leq 3.2\varepsilon_n + \sqrt{2\varepsilon_n |T_n|/n} + 1.5\sqrt{2\varepsilon_n \rho(h_{n+1}, h^*)}$$

where ε_n is defined in Eq. (5.4) *(with $\eta = \delta_n$).*

Proof. Lemma 5.3 implies that

$$\begin{aligned}\Delta(h_{n+1}, \tilde{S}_n, T_n, \delta_n) &= \varepsilon_n + \sqrt{2\varepsilon_n \cdot \frac{1}{n} \cdot \left(|T_n| + \sum_{(X_i, \tilde{Y}_i) \in \tilde{S}_n} \mathbb{1}(h(X_i) \neq \tilde{h}^*(X_i))\right)} \\ &\leq \varepsilon_n + \sqrt{2\varepsilon_n |T_n|/n} + \sqrt{2\varepsilon_n \rho_n(h_{n+1}, h^*)}\end{aligned}$$

Now applying the deviation bound from Eq. (5.2) to $\rho_n(h_{n+1}, h^*)$ and simplifying (using $2\sqrt{xy} \leq x + y$) gives the claim. □

Lemma 5.5. *Assume the conditions from Lemma 5.4. There exists a universal constant $C \in (0, 27)$ such that the following holds. Let $Q_{n+1} \in \{0, 1\}$ be the random variable that indicates if Y_{n+1} is queried. For all $n \geq 1$,*

$$\mathbb{E}[Q_{n+1}] \leq \theta \cdot \left((2 + \lambda) \cdot \mathrm{err}(h^*) + C \cdot \left(1 + \frac{1}{\lambda}\right) \cdot \varepsilon_n + 3.2 \cdot \sqrt{\varepsilon_n \cdot \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}[Q_n]}\right)$$

for all $\lambda > 0$.

Proof. Let

$$h := \begin{cases} h_{n+1} & \text{if } h'_{n+1}(X_{n+1}) = h^*(X_{n+1}) \\ h'_{n+1} & \text{if } h_{n+1}(X_{n+1}) = h^*(X_{n+1}) \end{cases}$$

so $h(X_{n+1}) \neq h^*(X_{n+1})$. Suppose Y_{n+1} is queried ($Q_{n+1} = 1$). We consider two possible cases:

1. If $h = h_{n+1}$, then

$$\begin{aligned}\mathrm{err}(h) - \mathrm{err}(h^*) &= \mathrm{err}(h_{n+1}) - \mathrm{err}(h^*) \\ &\leq \mathrm{err}(h_{n+1}, Z_n) - \mathrm{err}(h^*, Z_n) + \varepsilon_n + \sqrt{2\varepsilon_n \rho(h, h^*)} \\ &\leq \mathrm{err}(h_{n+1}, Z_n) - \mathrm{err}(h'_{n+1}, Z_n) + \varepsilon_n + \sqrt{2\varepsilon_n \rho(h, h^*)} \\ &\leq \varepsilon_n + \sqrt{2\varepsilon_n \rho(h, h^*)}\end{aligned}$$

   where the first inequality follows from Eq. (5.1), and the second follows from the fact $h'_{n+1}(X_{n+1}) = h^*(X_{n+1})$ and the definition of h'_{n+1}.

2. If instead $h = h'_{n+1}$, then

$$\begin{aligned}\mathrm{err}(h) - \mathrm{err}(h^*) &= \mathrm{err}(h'_{n+1}) - \mathrm{err}(h^*)\\ &\leq \mathrm{err}(h'_{n+1}, Z_n) - \mathrm{err}(h^*, Z_n) + \varepsilon_n + \sqrt{2\varepsilon_n \rho(h, h^*)}\\ &\leq \mathrm{err}(h'_{n+1}, Z_n) - \mathrm{err}(h_{n+1}, Z_n) + \varepsilon_n + \sqrt{2\varepsilon_n \rho(h, h^*)}\\ &\leq \Delta(h, \tilde{S}_n, T_n, \delta_n) + \varepsilon_n + \sqrt{2\varepsilon_n \rho(h, h^*)}\\ &\leq 4.2\varepsilon_n + 2.5\sqrt{2\varepsilon_n \rho(h, h^*)} + \sqrt{2\varepsilon_n |T_n|/n}\end{aligned}$$

   where the last inequality follows from Lemma 5.4.

In either case, we have by the triangle inequality,

$$\rho(h, h^*) \leq 2\,\mathrm{err}(h^*) + 4.2\varepsilon_n + 2.5\sqrt{2\varepsilon_n \rho(h, h^*)} + \sqrt{2\varepsilon_n |T_n|/n}.$$

Solving the quadratic inequality for $\rho(h, h^*)$ and simplifying gives

$$\rho(h, h^*) \leq (2 + \lambda) \cdot \mathrm{err}(h^*) + C \cdot \left(1 + \frac{1}{\lambda}\right) \cdot \varepsilon_n + 3.2 \cdot \sqrt{\varepsilon_n \cdot \frac{|T_n|}{n}} =: r_n$$

for any $\lambda > 0$. Therefore $X_{n+1} \in \mathcal{R}(h^*, r_n)$. Now using the definition of the disagreement coefficient θ,

$$\begin{aligned}\mathbb{E}[Q_{n+1}] &= \mathbb{E}[\mathbb{E}[Q_{n+1} | Z_n, X_{n+1}]]\\ &\leq \mathbb{E}[\Pr(X_{n+1} \in \mathcal{R}(h^*, r_n))]\\ &\leq \mathbb{E}[\theta \cdot r_n]\\ &= \theta \cdot \left((2 + \lambda) \cdot \mathrm{err}(h^*) + C \cdot \left(1 + \frac{1}{\lambda}\right) \cdot \varepsilon_n + \mathbb{E}\left[3.2 \cdot \sqrt{\varepsilon_n \cdot \frac{1}{n}\sum_{i=1}^{n} Q_n}\right]\right)\\ &\leq \theta \cdot \left((2 + \lambda) \cdot \mathrm{err}(h^*) + C \cdot \left(1 + \frac{1}{\lambda}\right) \cdot \varepsilon_n + 3.2 \cdot \sqrt{\varepsilon_n \cdot \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}[Q_n]}\right)\end{aligned}$$

where the last two steps use linearity of expectation and Jensen's inequality. □

Theorem 5.2. *There exists a universal constant $C > 0$ such that the following holds. Conditioned on an event that occurs with probability at least $1 - \delta$, the expected number of labels queried by Oracular CAL after n iterations is at most*

$$1 + 2 \cdot \theta \cdot \mathrm{err}(h^*) \cdot (n - 1) + C \cdot \theta^2 \cdot \left(d + \log\frac{1}{\delta}\right) \cdot \ln^3 n \\ + C \cdot \theta^{3/2} \cdot \sqrt{\left(d + \log\frac{1}{\delta}\right) \cdot \mathrm{err}(h^*) \cdot (n - 1) \cdot \ln^3 n}.$$

Proof. Assuming Y_1 is always queried; applying Lemmas 5.1 and 5.5, and linearity of expectation; and optimizing over λ gives the bound

$$\begin{aligned}\sum_{i=1}^{n} \mathbb{E}[Q_i] \leq 1 + 2 \cdot \theta \cdot \operatorname{err}(h^*) \cdot (n-1) + C \cdot \int_1^n \varepsilon_x dx \\ + 2 \cdot \theta \cdot \sqrt{\operatorname{err}(h^*) \cdot (n-1) \cdot C \cdot \int_1^n \varepsilon_x dx} \\ + 3.2 \cdot \theta \cdot \int_1^n \sqrt{\frac{\varepsilon_x}{x}} dx \cdot \sqrt{\sum_{i=1}^{n} \mathbb{E}[Q_i]}.\end{aligned}$$

Evaluating the integrals and solving the quadratic for $\mathbb{E}[Q_1] + \ldots + \mathbb{E}[Q_n]$ completes the proof. □

The bound given here is worse than the bound for Agnostic CAL (Theorem 4.3) in its dependence on the disagreement coefficient θ in the sub-linear terms, but still implies a label complexity bound that improves over that of A^2 (Theorem 2.3).

5.2.4 Discussion

Comparing Agnostic CAL and Oracular CAL

The label complexity bound we derived for Oracular CAL appears to be weaker than that of Agnostic CAL. There are two possible avenues of improvement:

1. Tighten the analysis.
2. Use a different threshold function Δ.

The latter option can be carried out to some degree. In the $(n+1)$th iteration, the basic mechanism of inferring the label assigned by the optimal hypothesis h^* can be applied to every data point in T_n, in addition to the current point X_{n+1} (Lemma 5.3). Whenever it is possible to infer $h^*(X_i)$ for some $(X_i, Y_i) \in T_n$, the example is removed from T_n and placed in $\tilde{S}_n$, using the label $\tilde{Y}_i := h^*(X_i)$. This has the effect of lessening effect of the over-approximation

$$\sum_{(X_i,Y_i)\in T_n} \mathbb{1}(h(X_i) \neq h^*(X_i)) \leq |T_n|$$

used in the threshold function Δ. Unfortunately, carrying out this improvement does not seem to reduce the label complexity to that of Agnostic CAL. Thus, it seems that Oracular CAL pays a price for abandoning the strict version space approach, at least relative to Agnostic CAL.

On the other hand, Oracular CAL has qualitative advantages over Agnostic CAL that may be important in practice. The first is that tweaking Δ to be more aggressive has less severe consequences in Oracular CAL than in Agnostic CAL. That is, the failure mode of Oracular CAL is that it sometimes sets $\tilde{Y}_i \neq h^*(X_i)$, which seems fine as long as it doesn't happen too often. In contrast, if Agnostic CAL sets $\tilde{Y}_i \neq h^*(X_i)$, then h^* is evicted from the implicit version space, which can be devastating.

The second advantage is the computational advantage of the restricted use of the LEARN$_{\mathcal{H}}$ subroutine, a clear practical improvement.

Favorable Bias

Both Agnostic CAL and Oracular CAL ultimately create a labeled data set $\tilde{Z}_n = \tilde{S}_n \cup T_n$ that is biased. The bias is *favorable* in that it only makes h^* more attractive to a learner—this idea is formally expressed in Lemma 5.2.

However, sometimes even such a favorable bias can be undesirable. For instance, the empirical error of a hypothesis computed on $\tilde{Z}_n$ is no longer an unbiased estimator of its true error; this cannot even be compensated for with uniform deviations bounds. Moreover, the bias may drop out of favor if h^* is no longer sought after, *e.g.*, if the hypothesis class changes.

5.3 An Importance Weighting Algorithm

We now describe an algorithm that overcomes the issue of creating a biased data set. In fact, the algorithm will avoid synthesizing labels altogether, and instead add *importance weights* to data for which labels are queried. For an importance weighted set of examples $S \subset \mathcal{X} \times \mathcal{Y} \times \mathbb{R}_+$, the *importance weighted empirical error* of a hypothesis $h : \mathcal{X} \to \mathcal{Y}$ is

$$\operatorname{err}(h, S, m) := \frac{1}{m} \sum_{(x,y,w) \in S} w \cdot \mathbb{1}(h(x) \neq y) \tag{5.8}$$

(m is a suitable normalizing constant). The weights will be set in such a way that guarantees $\mathbb{E}[\operatorname{err}(h, S, m)] = \operatorname{err}(h)$. Here, the expectation includes the internal randomness used by the algorithm in forming the weighted set of examples S. The primary challenge, then, will be in controlling the variance of these estimates.

The algorithm (Algorithm 5.2) is based on the Importance Weighted Active Learning (IWAL) framework of [BDL09]; we call our particular instantiation IWAL-CAL, as it combines a technique from Oracular CAL with the importance weighting trick. Note that, like Oracular CAL, IWAL-CAL only requires the enforcement of a single hard constraint for determining h'_t, and does not require any hard constraints for determining h_t. However, it does require the minimization of an importance weighted empirical error, which may add some computational complexity.

As we did for Oracular CAL, we will assume for simplicity that the entire hypothesis class $\mathcal{H}$ does not completely agree on any single data point $x \in \mathcal{X}$. That is, for each $x \in \mathcal{X}$, there exists $h, h' \in \mathcal{H}$ such that $h(x) = 1$ and $h'(x) = -1$. If the learner should come across any points for which this assumption fails, it can choose any query probability $P_t \in (0, 1]$ (*e.g.*, $P_t = 1/t$) without affecting the behavior of the algorithm with respect to the rest of the data points. We will also assume for simplicity that $\mathcal{H}$ is finite. This can be relaxed by letting $\mathcal{H}$ be a finite ϵ-cover of an infinite class.

```
Algorithm 5.2 (IWAL-CAL)
Notes: see Eq. (5.8) for the definition of err (importance weighted error), and Section 5.3.4 for the definitions of $C_0$, $c_1$, and $c_2$.
Initialize: $S_0 := \emptyset$.
For $t = 1, 2, \ldots, n$:
    1. Obtained unlabeled data point $X_t$.
    2. Let
        (a) $h_t := \arg\min\{\mathrm{err}(h, S_{t-1}, t-1) : h \in \mathcal{H}\}$, and
        (b) $h'_t := \arg\min\{\mathrm{err}(h, S_{t-1}, t-1) : h \in \mathcal{H} \ \wedge \ h(X_t) \neq h_t(X_t)\}$.
    3. Let $G_t := \mathrm{err}(h'_t, S_{t-1}, t-1) - \mathrm{err}(h_t, S_{t-1}, t-1)$, and
        $$P_t := \begin{cases} 1 & \text{if } G_t \leq \sqrt{\frac{C_0 \log t}{t-1}} + \frac{C_0 \log t}{t-1} \\ s(G_t, t) & \text{otherwise} \end{cases}$$
       where $s(g, t) \in (0, 1)$ is the positive solution $s$ to the equation
        $$g = \left( \frac{c_1}{\sqrt{s}} - c_1 + 1 \right) \cdot \sqrt{\frac{C_0 \log t}{t-1}} + \left( \frac{c_2}{s} - c_2 + 1 \right) \cdot \frac{C_0 \log t}{t-1}.$$
    4. Toss a biased coin with $\Pr(\text{heads}) = P_t$.
        (a) If heads, then query $Y_t$, and let $S_t := S_{t-1} \cup \{(X_t, Y_t, 1/P_t)\}$.
        (b) Else, let $S_t := S_{t-1}$.
Return: $h_{n+1} := \arg\min\{\mathrm{err}(h, S_n) : h \in \mathcal{H}\}$.
```

Figure 5.2: The IWAL-CAL algorithm.

In the remainder of this chapter, we will use the notation $a_{1:n}$ to denote a sequence $(a_1, a_2, \ldots, a_n)$.

5.3.1 Importance Weighted Active Learning

In the IWAL framework, the learner chooses a query probability $P_t \in (0, 1]$ after receiving each new point X_t. Then, a coin with heads bias P_t is tossed; the label Y_t is queried if the coin comes up heads, and otherwise the label is foregone. The query probability P_t can depend on all previous unlabeled examples, any previously queried labels, the outcomes of any of the past coin tosses, and the current unlabeled point X_t.

Formally, an IWAL algorithm specifies a rejection threshold function $p : (\mathcal{X} \times \mathcal{Y} \times \{0, 1\})^* \times \mathcal{X} \to (0, 1]$ for determining these query probabilities. Let $Q_t \in \{0, 1\}$ be a random variable conditionally independent of the current label Y_t

$$Q_t \perp\!\!\!\perp Y_t \mid X_{1:t}, Y_{1:t-1}, Q_{1:t-1}$$

and with conditional expectation

$$\mathbb{E}[Q_t | X_{1:t}, Y_{1:t-1}, Q_{1:t-1}] \;=\; P_t \;:=\; p(Z_{1:t-1}, X_t).$$

where

$$Z_t \;:=\; (X_t, Y_t, Q_t).$$

That is, Q_t indicates if the label Y_t is queried (the outcome of the coin toss). The query probability P_t is allowed to depend on a label Y_t if and only if it has been queried, *i.e.*, iff the corresponding $Q_t = 1$.

5.3.2 Importance Weighted Estimators

Let $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ be a function over $\mathcal{X} \times \mathcal{Y}$. The *importance weighted estimator* of $\mathbb{E}[f(X, Y)]$ from $Z_{1:n} \in (\mathcal{X} \times \mathcal{Y} \times \{0, 1\})^n$ is

$$\widehat{f}(Z_{1:n}) \;:=\; \frac{1}{n} \sum_{i=1}^{n} \frac{Q_i}{P_i} \cdot f(X_i, Y_i).$$

Note that this quantity depends on a label Y_i only if it has been queried (*i.e.*, only if $Q_i = 1$; it also depends on X_i only if $Q_i = 1$). The IWAL-CAL algorithm uses a rejection threshold function solely based on estimators of this type.

A basic property of the importance weighted estimator $\widehat{f}$ is *unbiasedness*:

$$\begin{aligned}
\mathbb{E}[\widehat{f}(Z_{1:n})] &= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\frac{Q_i\cdot f(X_i,Y_i)}{P_i}\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\mathbb{E}\left[\left.\frac{Q_i\cdot f(X_i,Y_i)}{P_i}\right| X_{1:i},Y_{1:i},Q_{1:i-1}\right]\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\frac{\mathbb{E}\left[Q_i|\, X_{1:i},Y_{1:i},Q_{1:i-1}\right]}{P_i}\cdot f(X_i,Y_i)\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\frac{\mathbb{E}\left[Q_i|\, X_{1:i},Y_{1:i-1},Q_{1:i-1}\right]}{P_i}\cdot f(X_i,Y_i)\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\frac{P_i}{P_i}\cdot f(X_i,Y_i)\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[f(X_i,Y_i)] \\
&= \mathbb{E}[f(X,Y)].
\end{aligned}$$

For instance, an unbiased estimator of the error of a hypothesis $h:\mathcal{X}\to\mathcal{Y}$ is

$$\operatorname{err}(h,Z_{1:n}) \;:=\; \frac{1}{n}\sum_{i=1}^{n}\frac{Q_i}{P_i}\cdot\mathbb{1}(h(X_i)\neq Y_i).$$

In the notation of Algorithm 5.2 and Eq. (5.8), this is equivalent to $\operatorname{err}(h,S_n,n)$, where

$$S_n \;:=\; \{(X_i,Y_i,1/P_i) : 1\le i\le n \;\wedge\; Q_i=1\}$$

is the importance weighted data set collected by IWAL-CAL.

5.3.3 A Deviation Bound for Importance Weighted Estimators

As mentioned before, the rejection threshold used by IWAL-CAL is based on importance weighted error estimates $\operatorname{err}(h,Z_{1:n})$. Even though these estimates are unbiased, they are only reliable when the variance is not too large. To get a handle on this, we need a deviation bound for importance weighted estimators. This is complicated by two factors:

1. The importance weighted samples $(X_i,Y_i,1/P_i)$ (or equivalently, the $Z_i=(X_i,Y_i,Q_i)$) are not iid. This is because the query probability P_i (and thus the importance weight $1/P_i$) generally depends on $Z_{1:i-1}$ and X_i.
2. The effective range of each term in the estimator is, itself, a random variable.

To address these issues, we develop a deviation bound based on a martingale technique from [Zha05].

Let $f:\mathcal{X}\times\mathcal{Y}\to[-1,1]$ be a bounded function. Consider any rejection threshold function $p:(\mathcal{X}\times\mathcal{Y}\times\{0,1\})^*\times\mathcal{X}\to(0,1]$ for which $P_n=p(Z_{1:n-1},X_n)$ is bounded below by some quantity (which may depend on n). Equivalently, the query probabilities P_n should have inverses $1/P_n$ bounded above by some r_{max} (which, again, may depend on n). The *a priori* upper bound r_{max} on $1/P_n$ can be pessimistic, as the dependence on r_{max} in the final deviation bound will be very mild: it enters in as $\log\log r_{max}$.

Let

$$W_i := \frac{Q_i}{P_i} \cdot f(X_i, Y_i)$$

be the ith term in the importance weighted estimator

$$\widehat{f}(Z_{1:n}) := \frac{1}{n}\sum_{i=1}^{n} W_i.$$

Our goal is to prove a bound on $|\widehat{f}(Z_{1:n}) - \mathbb{E}[f(X,Y)]|$ that holds with high probability over the joint distribution of $Z_{1:n}$.

To start, we establish bounds on the range and variance of each term W_i in the estimator, conditioned on $(X_{1:i}, Y_{1:i}, Q_{1:i-1})$. Write $\mathbb{E}_i[\,\cdot\,]$ to denote $\mathbb{E}[\,\cdot\,|X_{1:i}, Y_{1:i}, Q_{1:i-1}]$. Note that

$$\mathbb{E}_i[W_i] \;=\; \frac{\mathbb{E}_i[Q_i]}{P_i} \cdot f(X_i, Y_i) \;=\; \frac{P_i}{P_i} \cdot f(X_i, Y_i) \;=\; f(X_i, Y_i) \tag{5.9}$$

so if $\mathbb{E}_i[W_i] = 0$, then $W_i = 0$. Therefore, the (conditional) range and variance are non-zero only if $\mathbb{E}_i[W_i] \neq 0$. For the range, we have

$$|W_i| \;=\; \frac{|Q_i|}{P_i} \cdot |f(X_i, Y_i)| \;\leq\; \frac{1}{P_i} \tag{5.10}$$

and, for the variance,

$$\mathbb{E}_i[(W_i - \mathbb{E}_i[W_i])^2] \;\leq\; \frac{\mathbb{E}_i[Q_i^2]}{P_i^2} \cdot f(X_i, Y_i)^2 \;=\; \frac{P_i}{P_i^2} \cdot f(X_i, Y_i)^2 \;\leq\; \frac{1}{P_i}. \tag{5.11}$$

Our approach is as follows. First, we show via a martingale inequality that

$$\left| \frac{1}{n}\sum_{i=1}^{n} W_i - \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_i[W_i] \right| \;\leq\; O\left(\sqrt{\frac{1}{P_{min}} \cdot \frac{\log\log r_{max}}{n}} + \frac{1}{P_{min}} \cdot \frac{\log\log r_{max}}{n} \right)$$

with high probability, where $P_{min} := \min\{P_i : 1 \leq i \leq n \;\wedge\; \mathbb{E}_i[W_i] \neq 0\}$ and $1/P_{min} \leq r_{max}$. As $\mathbb{E}_i[W_i] = f(X_i, Y_i)$, this is a bound on the difference between the importance-weighted estimator and the fully-supervised estimator. Next, we use Hoeffding's inequality to get

$$\left| \frac{1}{n}\sum_{i=1}^{n} f(X_i, Y_i) - \mathbb{E}[f(X,Y)] \right| \;\leq\; O\left(\sqrt{\frac{1}{n}} \right)$$

with high probability. Finally, we combine the two bounds with the triangle inequality.

The techniques here are mostly developed in [Zha05]; for completeness, we detail the proofs for our particular application. The first two lemmas establish a basic bound in terms of conditional moment generating functions.

Lemma 5.6. *For all $n \geq 1$ and all functionals $\Xi_i := \xi_i(Z_{1:i})$,*

$$\mathbb{E}\left[\exp\left(\sum_{i=1}^{n}\Xi_i - \sum_{i=1}^{n}\ln\mathbb{E}_i[\exp(\Xi_i)]\right)\right] \;=\; 1.$$

Proof. A straightforward induction on n. □

Lemma 5.7. *For all $t \geq 0$, $\lambda \in \mathbb{R}$, $n \geq 1$, and functionals $\Xi_i := \xi_i(Z_{1:i})$,*

$$\Pr\left(\lambda\sum_{i=1}^{n}\Xi_i - \sum_{i=1}^{n}\ln\mathbb{E}_i[\exp(\lambda\Xi_i)] \geq t\right) \leq e^{-t}.$$

Proof. The claim follows by Markov's inequality and Lemma 5.6 (replacing Ξ_i with $\lambda\Xi_i$). □

In order to specialize Lemma 5.7 for our purposes, we first analyze the conditional moment generating function of $W_i - \mathbb{E}_i[W_i]$.

Lemma 5.8. *If $0 < \lambda < 3P_i$, then*

$$\ln\mathbb{E}_i[\exp(\lambda(W_i - \mathbb{E}_i[W_i]))] \;\leq\; \frac{1}{P_i}\cdot\frac{\lambda^2}{2(1-\lambda/(3P_i))}.$$

If $\mathbb{E}_i[W_i] = 0$, then

$$\ln\mathbb{E}_i[\exp(\lambda(W_i - \mathbb{E}_i[W_i]))] \;=\; 0.$$

Proof. Let $g(x) := (\exp(x) - x - 1)/x^2$ for $x \neq 0$, so $\exp(x) = 1 + x + x^2 \cdot g(x)$. Note that $g(x)$ is non-decreasing. Thus,

$$\begin{aligned}
&\mathbb{E}_i\left[\exp(\lambda(W_i - \mathbb{E}_i[W_i]))\right]\\
&\quad= \mathbb{E}_i\left[1 + \lambda(W_i - \mathbb{E}_i[W_i]) + \lambda^2(W_i - \mathbb{E}_i[W_i])^2 \cdot g(\lambda(W_i - \mathbb{E}_i[W_i]))\right]\\
&\quad= 1 + \lambda^2 \cdot \mathbb{E}_i\left[(W_i - \mathbb{E}_i[W_i])^2 \cdot g(\lambda(W_i - \mathbb{E}_i[W_i]))\right]\\
&\quad\leq 1 + \lambda^2 \cdot \mathbb{E}_i\left[(W_i - \mathbb{E}_i[W_i])^2 \cdot g(\lambda/P_i)\right]\\
&\quad= 1 + \lambda^2 \cdot \mathbb{E}_i\left[(W_i - \mathbb{E}_i[W_i])^2\right] \cdot g(\lambda/P_i)\\
&\quad\leq 1 + (\lambda^2/P_i) \cdot g(\lambda/P_i)
\end{aligned}$$

where the first inequality follows from the range bound (Eq. (5.10)) and the second follows from variance bound (Eq. (5.11)). Now the first claim follows from the definition of $g(x)$, the facts $\exp(x) - x - 1 \leq x^2/(2(1 - x/3))$ for $0 \leq x < 3$ and $\ln(1 + x) \leq x$.

The second claim is immediate from the facts $\mathbb{E}_i[W_i] = f(X_i, Y_i)$ (Eq. (5.9)) and $W_i = (Q_i/P_i) \cdot f(X_i, Y_i)$. □

We now combine Lemma 5.8 and Lemma 5.7 to bound the deviation of the importance weighted estimator $\widehat{f}(Z_{1:n})$ from $(1/n)\sum_{i=1}^{n}\mathbb{E}_i[W_i]$.

Lemma 5.9. *Pick any $t \geq 0$, $n \geq 1$, and $p_{min} > 0$, and let E be the (joint) event*

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n}W_i - \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_i[W_i] \;&\geq\; \sqrt{\frac{1}{p_{min}}\cdot\frac{2t}{n}} + \frac{1}{p_{min}}\cdot\frac{t}{3n}\\
\textit{and}\quad \min\{P_i : 1 \leq i \leq n \;\wedge\; \mathbb{E}_i[W_i] \neq 0\} \;&\geq\; p_{min}.
\end{aligned}$$

Then $\Pr(E) \leq e^{-t}$.

Proof. With foresight, let

$$\lambda \;:=\; 3p_{min} \cdot \frac{\sqrt{\frac{1}{3p_{min}} \cdot \frac{2t}{3n}}}{1 + \sqrt{\frac{1}{3p_{min}} \cdot \frac{2t}{3n}}}.$$

Note that $0 < \lambda < 3p_{min}$. By Lemma 5.8 and the choice of λ, we have that if $\min\{P_i : 1 \leq i \leq n \;\wedge\; \mathbb{E}_i[W_i] \neq 0\} \geq p_{min}$, then

$$\frac{1}{n\lambda} \cdot \sum_{i=1}^{n} \ln \mathbb{E}_i[\exp(\lambda(W_i - \mathbb{E}_i[W_i]))] \;\leq\; \frac{1}{p_{min}} \cdot \frac{\lambda}{2(1-\lambda/(3p_{min}))} \;=\; \sqrt{\frac{1}{p_{min}} \cdot \frac{t}{2n}} \tag{5.12}$$

and

$$\frac{t}{n\lambda} \;=\; \sqrt{\frac{1}{p_{min}} \cdot \frac{t}{2n}} + \frac{1}{p_{min}} \cdot \frac{t}{3n}. \tag{5.13}$$

Let E' be the event that

$$\frac{1}{n} \cdot \sum_{i=1}^{n} (W_i - \mathbb{E}_i[W_i]) - \frac{1}{n\lambda} \cdot \sum_{i=1}^{n} \ln \mathbb{E}_i[\exp(\lambda(W_i - \mathbb{E}_i[W_i]))] \;\geq\; \frac{t}{n\lambda}$$

and let E'' be the event $\min\{P_i : 1 \leq i \leq n \;\wedge\; \mathbb{E}_i[W_i] \neq 0\} \;\geq\; p_{min}$. Together, Eq. (5.12) and Eq. (5.13) imply $E \subseteq E' \cap E''$. And of course, $E' \cap E'' \subseteq E'$, so $\Pr(E) \leq \Pr(E' \cap E'') \leq \Pr(E') \leq e^{-t}$ by Lemma 5.7. □

To do away with the joint event in Lemma 5.9, we use the standard trick of taking a union bound over a geometrical sequence of possible values for p_{min}.

Lemma 5.10. *Pick any $t \geq 0$ and $n \geq 1$. Assume $1 \leq 1/P_i \leq r_{max}$ for all $1 \leq i \leq n$, and let $R_n := 1/\min\{P_i : 1 \leq i \leq n \;\wedge\; \mathbb{E}_i[W_i] \neq 0\} \cup \{1\}$. We have*

$$\Pr\left(\left| \frac{1}{n} \sum_{i=1}^{n} W_i - \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_i[W_i] \right| \;\geq\; \sqrt{\frac{2R_n t}{n}} + \frac{R_n t}{3n} \right) \;\leq\; 2(2 + \log_2 r_{max}) e^{-t/2}.$$

Proof. The assumption on P_i implies $1 \leq R_n \leq r_{max}$. Let $r_j := 2^j$ for $-1 \leq j \leq m := \lceil \log_2 r_{max} \rceil$. Then

$$\begin{aligned}
&\Pr\left(\frac{1}{n} \sum_{i=1}^{n} W_i - \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_i[W_i] \;\geq\; \sqrt{\frac{2R_n t}{n}} + \frac{R_n t}{3n} \right) \\
&\quad= \sum_{j=0}^{m} \Pr\left(\frac{1}{n} \sum_{i=1}^{n} W_i - \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_i[W_i] \;\geq\; \sqrt{\frac{2R_n t}{n}} + \frac{R_n t}{3n} \;\wedge\; r_{j-1} < R_n \leq r_j \right) \\
&\quad\leq \sum_{j=0}^{m} \Pr\left(\frac{1}{n} \sum_{i=1}^{n} W_i - \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_i[W_i] \;\geq\; \sqrt{\frac{2r_{j-1} t}{n}} + \frac{r_{j-1} t}{3n} \;\wedge\; R_n \leq r_j \right) \\
&\quad= \sum_{j=0}^{m} \Pr\left(\frac{1}{n} \sum_{i=1}^{n} W_i - \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_i[W_i] \;\geq\; \sqrt{\frac{2r_j (t/2)}{n}} + \frac{r_j (t/2)}{3n} \;\wedge\; R_n \leq r_j \right) \\
&\quad\leq (2 + \log_2 r_{max}) e^{-t/2}
\end{aligned}$$

where the last inequality follows from Lemma 5.9. Replacing W_i with $-W_i$ bounds the probability of deviations in the other direction in exactly the same way. The claim then follows by the union bound. □

Finally, we bound the deviation of the supervised estimator from $\mathbb{E}[f(X, y)]$, and combine this with Lemma 5.10 to give our final deviation bound.

Theorem 5.3. *Pick any $t \geq 0$ and $n \geq 1$. Assume $1 \leq 1/P_i \leq r_{max}$ for all $1 \leq i \leq n$, and let $R_n := 1/\min\{P_i : 1 \leq i \leq n \ \wedge \ f(X_i, Y_i) \neq 0\} \cup \{1\}$. With probability at least $1 - 2(3 + \log_2 r_{max})e^{-t/2}$,*

$$\left| \frac{1}{n} \sum_{i=1}^{n} \frac{Q_i}{P_i} \cdot f(X_i, Y_i) - \mathbb{E}[f(X, Y)] \right| \ \leq \ \sqrt{\frac{2R_n t}{n}} + \sqrt{\frac{2t}{n}} + \frac{R_n t}{3n}.$$

Proof. By Hoeffding's inequality and the fact $|f(X_i, Y_i)| \leq 1$, we have

$$\Pr\left(\left| \frac{1}{n} \sum_{i=1}^{n} f(X_i, Y_i) - \mathbb{E}[f(X, Y)] \right| \ \geq \ \sqrt{\frac{2t}{n}} \right) \ \leq \ 2e^{-t/2}.$$

Since $\mathbb{E}_i[W_i] = f(X_i, Y_i)$, the claim follows by combining this and Lemma 5.10 with the triangle inequality and the union bound. □

5.3.4 The IWAL-CAL Rejection Threshold

First, we state a deviation bound for the importance weighted error of hypotheses in a finite hypothesis class $\mathcal{H}$ that holds for all $n \geq 1$. It is a simple consequence of Theorem 5.3 and union bounds.

Lemma 5.11. *Pick any $\delta \in (0, 1)$. For all $n \geq 1$, let*

$$\varepsilon_n := \frac{16 \log(2(3 + n \log_2 n) n(n+1)|\mathcal{H}|/\delta)}{n} = O\left(\frac{\log(n|\mathcal{H}|/\delta)}{n} \right). \tag{5.14}$$

Let $(Z_1, Z_2, \ldots) \in (\mathcal{X} \times \mathcal{Y} \times \{0,1\})^*$ be the sequence of random variables specified in Section 5.3.1 using a rejection threshold function $p : (\mathcal{X} \times \mathcal{Y} \times \{0,1\})^* \times \mathcal{X} \to [0, 1]$ that satisfies

$$p(z_{1:n}, x) \geq 1/n^n$$

for all $n \geq 1$ and all $(z_{1:n}, x) \in (\mathcal{X} \times \mathcal{Y} \times \{0,1\})^n \times \mathcal{X}$.

The following holds with probability at least $1 - \delta$. For all $n \geq 1$ and all $h \in \mathcal{H}$,

$$\begin{aligned} &|(\mathrm{err}(h, Z_{1:n}) - \mathrm{err}(h^*, Z_{1:n})) - (\mathrm{err}(h) - \mathrm{err}(h^*))| \\ &\leq \sqrt{\frac{1}{P_{min,n}(h)} \cdot \varepsilon_n} + \frac{1}{P_{min,n}(h)} \cdot \varepsilon_n \end{aligned} \tag{5.15}$$

where

$$P_{min,n}(h) = \min\{P_i : 1 \leq i \leq n \ \wedge \ h(X_i) \neq h^*(X_i)\} \cup \{1\}.$$

We let $C_0 \geq 2$ be a quantity such that ε_n (as defined in Eq. (5.14)) is bounded as

$$\varepsilon_n \leq \frac{C_0 \cdot \log(n+1)}{n}.$$

The following constants are used in the description of the IWAL-CAL rejection threshold and the subsequent analysis:

$$c_1 := 5 + 2\sqrt{2} \qquad c_2 := 5 \qquad c_3 := \max\left(\left(\frac{c_1+\sqrt{2}}{c_1-2}\right)^2, \frac{c_2+2}{c_2-2}\right)$$

$$c_4 := (c_1 + \sqrt{c_3})^2 \qquad c_5 := c_2 + c_3.$$

The rejection threshold (line 3 in Algorithm 5.2) is based on the deviation bound from Lemma 5.11. First, the importance weighted error minimizing hypothesis h_t and the "alternative" hypothesis h'_t are found, and their difference in importance weighted errors G_t is computed. If $G_t \leq \sqrt{(C_0 \log t)/(t-1)} + (C_0 \log t)/(t-1)$, then the query probability P_t is set to 1. Otherwise, P_t is set to the positive solution s to the quadratic equation

$$G_t = \left(\frac{c_1}{\sqrt{s}} - c_1 + 1\right) \cdot \sqrt{\frac{C_0 \log t}{t-1}} + \left(\frac{c_2}{s} - c_2 + 1\right) \cdot \frac{C_0 \log t}{t-1}. \tag{5.16}$$

It can be checked that $P_t \in (0, 1]$ and that P_t is non-increasing with G_t. It is also useful to note that $\log(t+1)/t$ is monotonically decreasing with $t \geq 0$ (we use the convention $\log(1)/0 = \infty$).

In order to apply Lemma 5.11 with the IWAL-CAL rejection threshold, we need to establish the (very crude) bound $P_t \geq 1/t^t$ for all t.

Lemma 5.12. *The IWAL-CAL rejection threshold satisfies*

$$p(z_{1:n}, x) \geq 1/n^n$$

for all $n \geq 1$ *and all* $(z_{1:n}, x) \in (\mathcal{X} \times \mathcal{Y} \times \{0,1\})^{n-1} \times \mathcal{X}$.

Proof. By induction on n. Trivial for $n = 1$ (since $p(\varepsilon, x) = 1$ for all $x \in \mathcal{X}$), so now assume as the inductive hypothesis $p_{n-1} = p(z_{1:n-2}, x) \geq 1/(n-1)^{n-1}$ for all $(z_{1:n-2}, x) \in (\mathcal{X} \times \mathcal{Y} \times \{0,1\})^{n-2} \times \mathcal{X}$. Fix any $(z_{1:n-1}, x) \in (\mathcal{X} \times \mathcal{Y} \times \{0,1\})^{n-1} \times \mathcal{X}$, and consider the error difference g_n used to determine $p_n = p(z_{1:n}, x)$. We only have to consider the case $g_n > \sqrt{(C_0 \log n)/(n-1)} + (C_0 \log n)/(n-1)$. By the inductive hypothesis, we have $g_n \leq 2(n-1)^{n-1}$. Let $C'_0 := C_0 \log n$. Solving the quadratic in Eq. (5.16) implies

$$\sqrt{p_n} \;=\; \frac{c_1 \cdot \sqrt{\frac{C'_0}{n-1}} + \sqrt{\frac{c_1^2 \cdot C'_0}{n-1} + 4 \cdot \left(g_n + (c_1 - 1) \cdot \sqrt{\frac{C'_0}{n-1}} + (c_2 - 1) \cdot \frac{C'_0}{n-1}\right) \cdot \frac{c_2 \cdot C'_0}{n-1}}}{2\left(g_n + (c_1 - 1) \cdot \sqrt{\frac{C'_0}{n-1}} + (c_2 - 1) \cdot \frac{C'_0}{n-1}\right)}$$

so, very loosely,

$$p_n \;>\; \frac{c_2 \cdot C'_0}{c_1 \cdot (n-1) \cdot g_n} \;\geq\; \frac{c_2 \cdot C'_0}{2c_1 \cdot (n-1) \cdot (n-1)^{n-1}} \;>\; \frac{1}{e(n-1)^n} \;\geq\; \frac{1}{n^n}$$

as required. □

5.3.5 Correctness Analysis

We first prove a consistency guarantee for IWAL-CAL that bounds the generalization error of the importance weighted empirical error minimizer. The proof actually establishes a lower bound on the query probabilities $P_i \geq 1/2$ for X_i such that $h_n(X_i) \neq h^*(X_i)$. This offers an intuitive characterization of the weighting landscape induced by the importance weights $1/P_i$.

Theorem 5.4. *The following holds with probability at least* $1-\delta$. *For any* $n \geq 1$,

$$0 \;\leq\; \mathrm{err}(h_n) - \mathrm{err}(h^*) \;\leq\; \mathrm{err}(h_n, Z_{1:n-1}) - \mathrm{err}(h^*, Z_{1:n-1}) + \sqrt{\frac{2C_0 \log n}{n-1}} + \frac{2C_0 \log n}{n-1}.$$

This implies, for all $n \geq 1$,

$$\mathrm{err}(h_n) \;\leq\; \mathrm{err}(h^*) + \sqrt{\frac{2C_0 \log n}{n-1}} + \frac{2C_0 \log n}{n-1}.$$

Proof. We condition on the $1-\delta$ probability event that the deviation bounds from Lemma 5.11 hold. The proof now proceeds by induction on n. The claim is trivially true for $n=1$. Now pick any $n \geq 2$ and assume as the inductive hypothesis that

$$0 \;\leq\; \mathrm{err}(h_\tau) - \mathrm{err}(h^*) \;\leq\; \mathrm{err}(h_\tau, Z_{1:\tau-1}) - \mathrm{err}(h^*, Z_{1:\tau-1}) + \sqrt{\frac{2C_0 \log \tau}{\tau-1}} + \frac{2C_0 \log \tau}{\tau-1}. \quad (5.17)$$

for all $1 \leq \tau \leq n-1$. We need to show Eq. (5.17) holds for $\tau = n$.

Let $P_{min} := \min\{P_i : 1 \leq i \leq n-1 \;\wedge\; h_n(X_i) \neq h^*(X_i)\} \cup \{1\}$. If $P_{min} \geq 1/2$, then Eq. (5.15) implies that Eq. (5.17) holds for $\tau = n$ as needed. So assume for sake of contradiction that $P_{min} < 1/2$, and let $n_0 := \max\{i \leq n-1 : P_i = P_{min} \wedge h_n(X_i) \neq h^*(X_i)\}$. By definition of P_{n_0}, we have

$$\begin{aligned}
&\mathrm{err}(h'_{n_0}, Z_{1:n_0-1}) - \mathrm{err}(h_{n_0}, Z_{1:n_0-1}) \\
&\quad= \left(\frac{c_1}{\sqrt{P_{min}}} - c_1 + 1\right) \cdot \sqrt{\frac{C_0 \log n_0}{n_0-1}} + \left(\frac{c_2}{P_{min}} - c_2 + 1\right) \cdot \frac{C_0 \log n_0}{n_0-1}.
\end{aligned}$$

Using this fact together with the inductive hypothesis, we have

$$\begin{aligned}
&\mathrm{err}(h'_{n_0}, Z_{1:n_0-1}) - \mathrm{err}(h^*, Z_{1:n_0-1}) \\
&\quad= \mathrm{err}(h'_{n_0}, Z_{1:n_0-1}) - \mathrm{err}(h_{n_0}, Z_{1:n_0-1}) + \mathrm{err}(h_{n_0}, Z_{1:n_0-1}) - \mathrm{err}(h^*, Z_{1:n_0-1}) \\
&\quad\geq \left(\frac{c_1}{\sqrt{P_{min}}} - c_1 + 1\right) \cdot \sqrt{\frac{C_0 \log n_0}{n_0-1}} + \left(\frac{c_2}{P_{min}} - c_2 + 1\right) \cdot \frac{C_0 \log n_0}{n_0-1} \\
&\qquad - \sqrt{\frac{2C_0 \log n_0}{n_0-1}} - \frac{2C_0 \log n_0}{n_0-1} \\
&\quad= \left(\frac{c_1}{\sqrt{P_{min}}} - c_1 + 1 - \sqrt{2}\right) \cdot \sqrt{\frac{C_0 \log n_0}{n_0-1}} + \left(\frac{c_2}{P_{min}} - c_2 - 1\right) \cdot \frac{C_0 \log n_0}{n_0-1} \qquad (5.18) \\
&\quad> (c_1 - 1) \cdot (\sqrt{2} - 1) \cdot \sqrt{\frac{C_0 \log n_0}{n_0-1}} + (c_2 - 1) \cdot \frac{C_0 \log n_0}{n_0-1}
\end{aligned}$$

where the last step uses the fact that $P_{min} < 1/2$. Since this final quantity is positive, we have $\mathrm{err}(h'_{n_0}, Z_{1:n_0-1}) > \mathrm{err}(h^*, Z_{1:n_0-1})$. By the definition of h'_{n_0}, this implies $h'_{n_0}(X_{n_0}) \neq h^*(X_{n_0})$. Therefore, $h_n(X_{n_0}) = h'_{n_0}(X_{n_0})$ so $\mathrm{err}(h_n, Z_{1:n_0-1}) \geq \mathrm{err}(h'_{n_0}, Z_{1:n_0-1})$. Using this fact, Eq. (5.15), and Eq. (5.18), we have

$$\begin{aligned}
&\mathrm{err}(h_n, Z_{1:n-1}) - \mathrm{err}(h^*, Z_{1:n-1}) \\
&\geq \;\; \mathrm{err}(h_n) - \mathrm{err}(h^*) - \sqrt{\frac{1}{P_{min}} \cdot \frac{C_0 \log n}{n-1}} - \frac{1}{P_{min}} \cdot \frac{C_0 \log n}{n-1} \\
&\geq \;\; \mathrm{err}(h_n, Z_{1:n_0-1}) - \mathrm{err}(h^*, Z_{1:n_0-1}) \\
&\qquad -2 \cdot \sqrt{\frac{1}{P_{min}} \cdot \frac{C_0 \log n_0}{n_0-1}} - 2 \cdot \frac{1}{P_{min}} \cdot \frac{C_0 \log n_0}{n_0-1} \\
&\geq \;\; \left(\frac{c_1 - 2}{\sqrt{P_{min}}} - c_1 + 1 - \sqrt{2}\right) \cdot \sqrt{\frac{C_0 \log n_0}{n_0-1}} + \left(\frac{c_2 - 2}{P_{min}} - c_2 - 1\right) \cdot \frac{C_0 \log n_0}{n_0-1} \\
&> \;\; \left((c_1 - 1) \cdot (\sqrt{2} - 1) - 2\sqrt{2}\right) \cdot \sqrt{\frac{C_0 \log n_0}{n_0-1}} + (c_2 - 5) \cdot \frac{C_0 \log n_0}{n_0-1}
\end{aligned}$$

where, again, the last step uses the fact that $P_{min} < 1/2$. This final quantity is non-negative, so we have the contradiction $\mathrm{err}(h_n, Z_{1:n-1}) > \mathrm{err}(h^*, Z_{1:n-1})$. □

5.3.6 Label Complexity Analysis

We now bound the number of labels requested by IWAL-CAL after n iterations. First, we establish a property about the query probabilities that relates error deviations (via P_{min}) to empirical error differences (via P_n). Both quantities play essential roles in bounding the label complexity through the disagreement metric structure around h^*.

Lemma 5.13. *Assume the bounds from Eq.* (5.15) *holds for all $h \in \mathcal{H}$ and $n \geq 1$. For any $n \geq 1$, we have $P_n \leq c_3 \cdot P_{min}$, where $P_{min} := \min(\{P_i : 1 \leq i \leq n-1 \wedge h(X_i) \neq h^*(X_i)\} \cup \{1\})$ and*

$$h := \begin{cases} h_n & \text{if } h'_n(X_n) = h^*(X_n) \\ h'_n & \text{if } h_n(X_n) = h^*(X_n). \end{cases}$$

Proof. Assume for sake of contradiction that $P_{min} < P_n/c_3 \leq 1/c_3$, and let $n_0 := \max\{i \leq n-1 : P_i = P_{min} \;\wedge\; h(X_i) \neq h^*(X_i)\}$. Then, an argument similar to that from Theorem 5.4 (with the fact $c_3 \geq 2$) implies

$$\begin{aligned}
&\mathrm{err}(h, Z_{1:n_0-1}) - \mathrm{err}(h^*, Z_{1:n_0-1}) \\
&\geq \;\; \left(\frac{c_1}{\sqrt{P_{min}}} - c_1 + 1 - \sqrt{2}\right) \cdot \sqrt{\frac{C_0 \log n_0}{n_0-1}} + \left(\frac{c_2}{P_{min}} - c_2 - 1\right) \cdot \frac{C_0 \log n_0}{n_0-1}.
\end{aligned}$$

Therefore (again, similar to the proof of Theorem 5.4),

$$\begin{aligned}
&\operatorname{err}(h, Z_{1:n-1}) - \operatorname{err}(h^*, Z_{1:n-1}) \\
\geq\ & \left(\frac{c_1 - 2}{\sqrt{P_{min}}} - c_1 + 1 - \sqrt{2}\right) \cdot \sqrt{\frac{C_0 \log n_0}{n_0 - 1}} + \left(\frac{c_2 - 2}{P_{min}} - c_2 - 1\right) \cdot \frac{C_0 \log n_0}{n_0 - 1} \\
\geq\ & \left(\frac{c_1 - 2}{\sqrt{P_{min}}} - c_1 + 1 - \sqrt{2}\right) \cdot \sqrt{\frac{C_0 \log n}{n - 1}} + \left(\frac{c_2 - 2}{P_{min}} - c_2 - 1\right) \cdot \frac{C_0 \log n}{n - 1} \qquad (5.19) \\
>\ & 0.
\end{aligned}$$

If $h = h_n$, then by the definition of h_n and h'_n,

$$\begin{aligned}
\operatorname{err}(h, Z_{1:n-1}) - \operatorname{err}(h^*, Z_{1:n-1}) &= \operatorname{err}(h_n, Z_{1:n-1}) - \operatorname{err}(h^*, Z_{1:n-1}) \\
&\leq \operatorname{err}(h_n, Z_{1:n-1}) - \operatorname{err}(h'_n, Z_{1:n-1}) \\
&\leq 0,
\end{aligned}$$

a contradiction of the lower bound from above. Otherwise $h = h'_n$, so the definitions of h_n, h'_n, and P_n imply that

$$\begin{aligned}
&\operatorname{err}(h, Z_{1:n-1}) - \operatorname{err}(h^*, Z_{1:n-1}) \\
=\ & \operatorname{err}(h'_n, Z_{1:n-1}) - \operatorname{err}(h^*, Z_{1:n-1}) \\
\leq\ & \operatorname{err}(h'_n, Z_{1:n-1}) - \operatorname{err}(h_n, Z_{1:n-1}) \\
=\ & \left(\frac{c_1}{\sqrt{P_n}} - c_1 + 1\right) \cdot \sqrt{\frac{C_0 \log n}{n - 1}} + \left(\frac{c_2}{P_n} - c_2 + 1\right) \cdot \frac{C_0 \log n}{n - 1}. \qquad (5.20)
\end{aligned}$$

Combining the lower bound in Eq. (5.19) and upper bound Eq. (5.20), and using the assumption $P_{min} < P_n/c_3$ gives

$$\left(\frac{c_1 - \sqrt{c_3}(c_1 - 2)}{\sqrt{P_n}} + \sqrt{2}\right) \cdot \sqrt{\frac{C_0 \log n}{n - 1}} + \left(\frac{c_2 - c_3(c_2 - 2)}{P_n} + 2\right) \cdot \frac{C_0 \log n}{n - 1} > 0.$$

But $P_n \leq 1$, so each of the parenthesized terms is non-positive. This is a contradiction. □

The next lemma bounds the probability of querying the label Y_n; this is subsequently used to establish the final bound on the expected number of labels queried.

Lemma 5.14. *Assume the bounds from Eq.* (5.15) *holds for all $h \in \mathcal{H}$ and $n \geq 1$. Let $\eta := \operatorname{err}(h^*)$. For any $n \geq 1$,*

$$\mathbb{E}[Q_n] \quad \leq \quad \theta \ \cdot \ \left(2\eta + \sqrt{6c_4 \cdot \frac{C_0 \log n}{n - 1}} + \left(1 + \frac{1}{2} \log \frac{1}{\frac{3}{2}c_4 \cdot \frac{C_0 \log n}{n-1}}\right) \cdot \frac{3}{2}c_5 \cdot \frac{C_0 \log n}{n - 1}\right)$$

for all $\lambda > 0$.

Proof. Define

$$h := \begin{cases} h_n & \text{if } h'_n(X_n) = h^*(X_n) \\ h'_n & \text{if } h_n(X_n) = h^*(X_n). \end{cases}$$

By Lemma 5.13, we have

$$\min\left(\{P_i : 1 \le i \le n-1 \ \wedge \ h(X_i) \ne h^*(X_i)\} \cup \{1\}\right) \ge P_n/c_3.$$

If $h = h'_n$, then by Eq. (5.15) and the definitions of h_n, h'_n, and P_n,

$$\begin{aligned}
&\mathrm{err}(h) - \mathrm{err}(h^*) \\
&\quad = \ \mathrm{err}(h'_n) - \mathrm{err}(h^*) \\
&\quad \le \ \mathrm{err}(h'_n, Z_{1:n-1}) - \mathrm{err}(h^*, Z_{1:n-1}) + \sqrt{\frac{c_3}{P_n} \cdot \frac{C_0 \log n}{n-1}} + \frac{c_3}{P_n} \cdot \frac{C_0 \log n}{n-1} \\
&\quad \le \ \mathrm{err}(h'_n, Z_{1:n-1}) - \mathrm{err}(h_n, Z_{1:n-1}) + \sqrt{\frac{c_3}{P_n} \cdot \frac{C_0 \log n}{n-1}} + \frac{c_3}{P_n} \cdot \frac{C_0 \log n}{n-1} \\
&\quad = \ \left(\frac{c_1 + \sqrt{c_3}}{\sqrt{P_n}} - c_1 + 1\right) \cdot \sqrt{\frac{C_0 \log n}{n-1}} + \left(\frac{c_2 + c_3}{P_n} - c_2 + 1\right) \cdot \frac{C_0 \log n}{n-1} \\
&\quad \le \ \sqrt{\frac{c_4}{P_n}} \cdot \sqrt{\frac{C_0 \log n}{n-1}} + \frac{c_5}{P_n} \cdot \frac{C_0 \log n}{n-1}
\end{aligned}$$

where the last inequality follows because $c_1 \ge 1$ and $c_2 \ge 1$. If instead $h = h_n$, then again using the definitions of h_n, h'_n, and P_n,

$$\begin{aligned}
&\mathrm{err}(h) - \mathrm{err}(h^*) \\
&\quad = \ \mathrm{err}(h_n) - \mathrm{err}(h^*) \\
&\quad \le \ \mathrm{err}(h_n, Z_{1:n-1}) - \mathrm{err}(h^*, Z_{1:n-1}) + \sqrt{\frac{c_3}{P_n} \cdot \frac{C_0 \log n}{n-1}} + \frac{c_3}{P_n} \cdot \frac{C_0 \log n}{n-1} \\
&\quad \le \ \mathrm{err}(h_n, Z_{1:n-1}) - \mathrm{err}(h'_n, Z_{1:n-1}) + \sqrt{\frac{c_3}{P_n} \cdot \frac{C_0 \log n}{n-1}} + \frac{c_3}{P_n} \cdot \frac{C_0 \log n}{n-1} \\
&\quad \le \ \sqrt{\frac{c_3}{P_n} \cdot \frac{C_0 \log n}{n-1}} + \frac{c_3}{P_n} \cdot \frac{C_0 \log n}{n-1} \\
&\quad \le \ \sqrt{\frac{c_4}{P_n}} \cdot \sqrt{\frac{C_0 \log n}{n-1}} + \frac{c_5}{P_n} \cdot \frac{C_0 \log n}{n-1}.
\end{aligned}$$

If $\mathrm{err}(h) - \mathrm{err}(h^*) = \gamma > 0$, then solving the above quadratic inequality for P_n gives the bound

$$P_n \ \le \ \frac{3}{2} \cdot \left(\frac{c_4}{\gamma^2} + \frac{c_5}{\gamma}\right) \cdot \frac{C_0 \log n}{n-1}.$$

If $\mathrm{err}(h) - \mathrm{err}(h^*) \le \gamma$, then by the triangle inequality we have

$$\rho(h^*, h) \ \le \ \mathrm{err}(h^*) + \mathrm{err}(h) \ \le \ 2\,\mathrm{err}(h^*) + \gamma$$

which in turn implies $X_n \in \mathcal{R}(h^*, 2\eta + \gamma)$. Note that $\Pr(X_n \in \mathcal{R}(h^*, 2\eta + \gamma)) \le \theta \cdot (2\eta + \gamma)$.

Fix $\gamma_0 > 0$. We have

$$\begin{aligned}
\mathbb{E}[Q_n] &= \Pr(\mathrm{err}(h) - \mathrm{err}(h^*) \leq \gamma_0) \cdot \mathbb{E}[Q_n | \,\mathrm{err}(h) - \mathrm{err}(h^*) \leq \gamma_0] \\
&\quad + \int_{\gamma_0}^{1} \frac{\partial \Pr(\mathrm{err}(h) - \mathrm{err}(h^*) \leq \gamma)}{\partial \gamma} \cdot \mathbb{E}[Q_n | \,\mathrm{err}(h) - \mathrm{err}(h^*) = \gamma] \cdot d\gamma \\
&\leq \theta \cdot (2\eta + \gamma_0) + \int_{\gamma_0}^{1} \theta \cdot \frac{3}{2} \cdot \left(\frac{c_4}{\gamma^2} + \frac{c_5}{\gamma}\right) \cdot \frac{C_0 \log n}{n-1} \cdot d\gamma \\
&\leq \theta \cdot \left(2\eta + \gamma_0 + \frac{3}{2} \cdot \frac{C_0 \log n}{n-1} \cdot \left(\frac{c_4}{\gamma_0} + c_5 \cdot \log \frac{1}{\gamma_0}\right)\right).
\end{aligned}$$

Optimizing with respect to γ_0 completes the proof. □

Theorem 5.5. *Conditioned on an event that occurs with probability at least $1 - \delta$, the expected number of labels queried by IWAL-CAL after n iterations is at most*

$$1 \; + \; 2 \; \cdot \; \theta \; \cdot \; \mathrm{err}(h^*) \; \cdot \; (n \; - \; 1) \; + \; \theta\sqrt{6 c_4 C_0 n \log n} \; + \; \theta \left(1 + \frac{1}{2} \log \frac{n}{C_0}\right) \frac{3}{2} c_5 \log^2 n.$$

Proof. Follows from assuming Y_1 is always queried; applying Lemmas 5.11, 5.14, and linearity of expectation. □

This label complexity bound has the same leading terms $1 + 2 \cdot \theta \cdot \mathrm{err}(h^*) \cdot (n-1)$ as that of Agnostic CAL; the remaining terms somewhat worse than that of Agnostic CAL, but are still sublinear.

5.3.7 Labeling Rates Under Low Noise Conditions

Some recent work on active learning has focused on improved label complexity under certain noise conditions [CN06, BBZ07, CN07, Han09, Kol09]. Specifically, it is assumed that there exists constants $\kappa > 0$ and $0 < \alpha \leq 1$ such that

$$\rho(h, h^*) \;\leq\; \kappa \cdot (\mathrm{err}(h) - \mathrm{err}(h^*))^{\alpha} \tag{5.21}$$

for all $h \in \mathcal{H}$. This is related to Tsybakov's low noise condition [Tsy04]. Essentially, this condition requires that low error hypotheses not be too far from the optimal hypothesis h^* under the disagreement metric. Under this condition, Lemma 5.14 can be improved.

In the remainder of this section, we assume that for some value of $\kappa > 0$ and $0 < \alpha \leq 1$, the condition in Eq. (5.21) holds for all $h \in \mathcal{H}$.

Lemma 5.15. *Assume the bounds from Eq.* (5.15) *hold for all $h \in \mathcal{H}$ and $n \geq 1$. For any $n \geq 1$,*

$$\mathbb{E}[Q_n] \;\leq\; \theta \cdot \kappa \cdot c_{\alpha} \cdot \left(\frac{C_0 \log n}{n-1}\right)^{\alpha/2}$$

where c_α is a constant that depends only on α.

Proof. For the most part, the proof is the same as that of Lemma 5.14, so we just show where the noise condition from Eq. (5.21) enters. For $\alpha < 1$,

$$\begin{aligned}
\mathbb{E}[Q_n] &= \Pr(\mathrm{err}(h) - \mathrm{err}(h^*) \leq \gamma_0) \cdot \mathbb{E}[Q_n | \, \mathrm{err}(h) - \mathrm{err}(h^*) \leq \gamma_0] \\
&\quad + \int_{\gamma_0}^{1} \frac{\partial \Pr(\mathrm{err}(h) - \mathrm{err}(h^*) \leq \gamma)}{\partial \gamma} \cdot \mathbb{E}[Q_n | \, \mathrm{err}(h) - \mathrm{err}(h^*) = \gamma] \cdot d\gamma \\
&\leq \theta\kappa\gamma_0^\alpha + \int_{\gamma_0}^{1} \frac{\theta\kappa}{\alpha} \cdot \frac{1}{\gamma^{1-\alpha}} \cdot \frac{3}{2} \cdot \left(\frac{c_4}{\gamma^2} + \frac{c_5}{\gamma}\right) \cdot \frac{C_0 \log n}{n-1} \cdot d\gamma \\
&\leq \theta\kappa\gamma_0^\alpha + \frac{3\theta\kappa}{2\alpha} \cdot \frac{C_0 \log n}{n-1} \cdot \left(\frac{c_4}{2-\alpha} \cdot \frac{1}{\gamma_0^{2-\alpha}} + \frac{c_5}{1-\alpha} \cdot \frac{1}{\gamma_0^{1-\alpha}}\right).
\end{aligned}$$

The case $\alpha = 1$ can be handled similarly. Optimizing over γ_0 completes the proof. □

This lemma immediately implies the following bound on the number of label queries, which is sublinear for all $0 < \alpha \leq 1$.

Theorem 5.6. *Conditioned on an event that occurs with probability at least $1 - \delta$, the expected number of labels queried by IWAL-CAL after n iterations is at most*

$$\theta \cdot \kappa \cdot c_\alpha \cdot (C_0 \log n)^{\alpha/2} \cdot n^{1-\alpha/2}$$

where c_α is a constant that depends only on α.

5.3.8 Discussion

In this chapter, we have demonstrated that the strict version space approach can be relaxed with two different methods. The first (Oracular CAL) achieves this by relying on a more pessimistic threshold function Δ, while the second (IWAL-CAL) uses importance weights. These algorithms have qualitative advantages over Agnostic CAL that may be useful in practice.

The work of [BDL09], which originally introduced the IWAL framework, also presents a rejection threshold method called loss-weighting based on differences in importance weighted errors. In fact, the method is more general in that it works for other loss functions such as logistic loss, which can be efficiently optimized in certain cases. However, loss-weighting is unsatisfactory in two ways. First, computing the query probabilities requires an optimization over a strictly defined version space (similar to that used in an algorithm studied by [Kol09]). Second, the label complexity bound established in [BDL09] actually only holds for a hypothesis selected from this version space, rather than from the entire hypothesis class. In comparison, IWAL-CAL only requires optimizations over the entire hypothesis class, and its performance guarantees avoid any reliance on a version space.

5.4 Bibliographic Notes

This chapter is based on unpublished joint work with Alina Beygelzimer, John Langford, and Tong Zhang. The dissertation author was the primary investigator and author of this material.

Chapter 6

Experimental Evaluation

We report empirical results of applying the IWAL-CAL algorithm from Chapter 5 to various classification tasks.

6.1 Introduction

We are interested in experimentally comparing the practical performance of active learning to that of passive learning. It has previously been reported that some active learning algorithms actually perform *worse* than their passive learning counterparts [Set09]. This can likely be attributed to the mismanagement of the sampling bias that active learning introduces [SVP06, DH08]. Therefore, active learning has represented a risky endeavor by machine learning practitioners, who may simply opt for the safer approach of passive learning. On the other hand, the active learning algorithms presented in the previous chapters come with safety guarantees which roughly state that the algorithms enjoy label complexity bounds comparable to those of their passive learning counterparts. This mitigates the risk of employing active learning to some degree. However, an experimental study is still needed because

1. the label complexity bounds of active learning hide constants and logarithmic factors, which may be significant in practice, and

2. the algorithms described are often not implemented exactly, due to computational and other practical limitations.

In this chapter, we describe experiments using two instantiations of the IWAL-CAL algorithm (Algorithm 5.2) which differ in their (approximate) implementation of the error minimization oracle. The first uses a soft-margin support vector machine, and the second uses a standard decision tree learning algorithm. We compare their performance to that of a passive learner using the same base learning methods.

6.2 Algorithms

As mentioned above, we used two simple base learning algorithms to approximately implement the required error minimization oracle used by IWAL-CAL. In both cases, the unlabeled data space $\mathcal{X}$ is a d-dimensional Euclidean space $\mathbb{R}^d$, so each $x \in \mathcal{X}$ is represented by d real-value features.

6.2.1 Soft-Margin Support Vector Machine

The first base learning algorithm trains a linear classifier h_w represented by a weight vector $w \in \mathbb{R}^d$ ($h_w(x) = 1$ iff $w^\top x \geq 0$) to optimize the soft-margin support vector machine (SVM) objective [CV95], suitably modified to handle importance weights. For $S \subseteq \mathbb{R}^d \times \{\pm 1\} \times \mathbb{R}_+$, we select w to minimize

$$\frac{1}{2} \cdot \|w\|_2^2 \ + \ \frac{1}{|S|} \sum_{(x,y,1/p) \in S} \frac{1}{p} \cdot \max\left(0,\ 1 - y \cdot x^\top w\right) \tag{6.1}$$

(the regularization parameter typically denoted by λ is fixed to 1). To enforce a single example constraint (x, y), we minimize the objective in Eq. (6.1) subject to the linear constraint $y \cdot x^\top w \geq 0$. We used a dual Gauss-Seidel method for solving the optimization problems [Zha02].

6.2.2 Decision Tree

The second base learning algorithm uses the `J48` decision tree learning algorithm implemented in the Weka v3.6.2 data mining software [HFH$^+$09], with all default parameters. `J48` is a Java implementation of the popular C4.5 procedure [Qui93] for growing (and pruning) a decision tree, and it readily accommodates importance weights. To enforce a single example constraint (x, y), we use a simple heuristic: we learn a decision tree using the `J48` algorithm as is, and then change the label of the leaf node that predicts on x to y.

6.2.3 Rejection Threshold

We used the IWAL-CAL rejection threshold described in Chapter 5 (with different constants and logarithmic factors) for setting the query probabilities for both the SVM and decision tree methods. Recall, given the hypotheses h_t and h'_t in iteration t, the query probability P_t is determined in the following manner. Let $G_t := \operatorname{err}(h'_t, S_{t-1}) - \operatorname{err}(h_t, S_{t-1})$ be the difference of empirical importance weighted errors. If $G_t \leq \sqrt{C_0/(t-1)} + C_0/(t-1)$, then the query probability P_t is set to 1. Otherwise, P_t is set to the positive solution s to the quadratic equation

$$G_t = \sqrt{\frac{1}{s} \cdot \frac{C_0}{t-1}} + \frac{1}{s} \cdot \frac{C_0}{t-1},$$

which is

$$s = \frac{C_0}{t-1} \cdot \left(\frac{1 + \sqrt{1 + 4G_t}}{2G_t} \right)^2 .$$

We still have to specify the bound constant C_0. Our theoretical analysis uses

$$C_0 = O\left(\log \frac{|\mathcal{H}| n}{\delta} \right)$$

where $\mathcal{H}$ is the hypothesis class, n is the total number of data, and δ is the confidence parameter. However, it is well-known that theoretical generalization bounds are loose in practice, even when they account for various data-dependent factors such as the margin [SFBL98]. Therefore, we use much more optimistic settings of C_0. Our results are reported using $C_0 = 1/4$ for the SVM method, and $C_0 = 8$ for the decision tree method. Of course, using significantly smaller settings of C_0 results in overly-aggressive algorithms, while using larger choices of C_0 results in overly-conservative algorithms. We found that our algorithm is somewhat robust to the setting of C_0, and a single setting of C_0 worked well across different data sets with different characteristics (except for the extrinsic dimension, which was fixed in the experiments). One might imagine tuning C_0 using held-out (labeled) data, but (i) it is not obvious how to do so, and (ii) held-out data may be costly to obtain anyway. Therefore, it is an important open problem to develop a practical method of parameter tuning in active learning.

6.3 Experimental Setup

6.3.1 Binary Classification Tasks

We used the following data sets for binary classification experiments.

1. ADULT [AN07]: income prediction ($>$ \$50000 vs $\leq$ \$50000) from census data. We randomly chose 4000 of the 48842 data for training, and used the rest for testing.

2. KDDCUP99 [AN07]: network intrusion detection ("bad" vs "good" connection) from network usage statistics and connection features. We randomly choose 5000 of the 4000000 data for training, and another 5000 for testing.

3. MNIST3v5 [LBBH98]: handwritten digit classification ("3" vs "5") from pixel intensity values. We randomly chose 4000 of the 11552 training images for training, and used all of the 1902 testing images for testing.

These data sets were selected because they roughly correspond to three different levels of achievable error rate, with ADULT having the highest error rate, followed by MNIST3v5, and then KDDCUP99. We reduced the dimension of each data set to 25 using PCA and randomized the order of the training data. For the SVM base learner, we also normalized the length of each data vector.

6.3.2 Multi-Class Classification Task

We also conducted a multi-class classification experiment using the entire MNIST data set (all 10 digits). We only used the decision tree base learner for this experiment, since it naturally accommodates multi-class. The "alternative hypothesis" h'_t is forced to disagree with h_t on x_t by changing the label of the leaf node that predicts on x_t to the next best label. For this experiment, the number of training data is 60000 and the number of testing data is 10000. We reduced the dimension of the data to 40 using PCA and randomized the order of the training data.

6.3.3 Evaluation Procedure

We compared the performance of IWAL-CAL to a passive learner using the same base learning algorithm. The passive learner can be thought of as an active learner that simply chooses unlabeled data at random to label; alternatively, in the "online" framework in which the unlabeled data arrive one at a time, the passive learner simply queries every label.

We consider two different error rates. First, we consider the test error after each learner has observed n unlabeled data. We call this the unlabeled error rate. Second, we consider the test error after each learner has queried n labels. We call this the labeled error rate. Note that the unlabeled and labeled error rates are the same for the passive learner.

6.4 Results

6.4.1 Binary Classification Experiments

The unlabeled error rates for both base learners are plotted in Figure 6.1. In most of the cases, the plots for active and passive learning are similar, which is in accord with the safety guarantee of IWAL-CAL—that the active learner enjoys roughly the same unlabeled error rate as a passive learner. The error rate somewhat more variable for the active learner, which may be due to the use of an importance weighted sample. As discussed in Chapter 5, the importance weighted sample provides unbiased estimates of the error, but the variance of these estimates are larger than the usual fully-supervised estimates.

The labeled error rates for both base learners are plotted in Figure 6.2, with close-ups in Figure 6.3. The active learner dramatically improves over the passive learner on the KDDCUP99 data set with both the base learners. The improvement is very modest on ADULT and MNIST3v5 using the decision tree base learner, and non-existent on ADULT and MNIST3v5 using the SVM base learner. Note that the error rate on KDDCUP99 is also very small, so the good performance there relative to the other two data sets is explained by the dependence on the noise rate in the label complexity bounds for IWAL-CAL (Theorems 5.5 and 5.6). It seems that the benefits of active learning are most apparent with learning problems with low levels of noise.

Finally, we also plot the labeling rates (the number of labels queried versus the number of unlabeled data seen) in Figure 6.4. These plots confirm the dramatic improvement of the active learner over the passive learner on KDDCUP99, where the labeling rate appears very sublinear, and the modest improvements on ADULT and MNIST3v5 using the decision tree base learner, where the labeling rate appears linear or only slightly sublinear. The labeling rate on ADULT and MNIST3v5 using the SVM base learner is almost exactly the same as that for passive learning, *i.e.*, the active learner queries almost every label. Therefore, the degraded (and more variable) test error may be due to the increased variance in the error estimates due to the non-unitary importance weights.

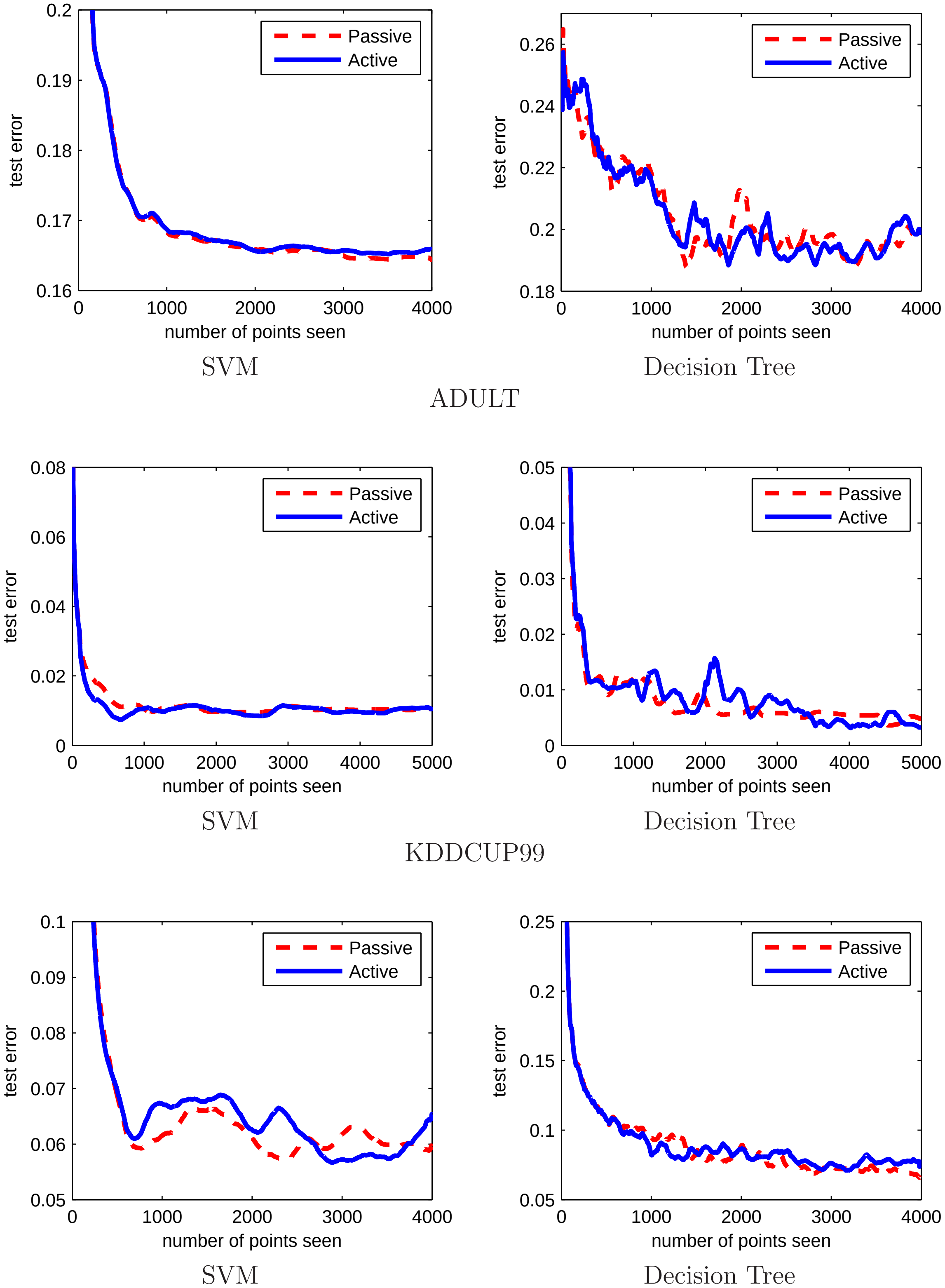
6.4.2 Multi-Class Classification Experiment

The plots for the multi-class classification experiment are in Figure 6.5. Here, we observe a modest improvement of the active learner over the passive learner in the labeled error rate.

The active learner queried a little over $^2/_3$ of the labels. While this may be seen as a sizable fraction of the data set, it does correspond to noticeable savings. For instance, the passive learner required over 6500 more label queries than the active learner to achieve an error rate of 0.175.

6.5 Bibliographic Notes

This chapter is based on unpublished joint work with Alina Beygelzimer, John Langford, and Tong Zhang. The dissertation author was the primary investigator and author of this material.



SVM Decision Tree

ADULT

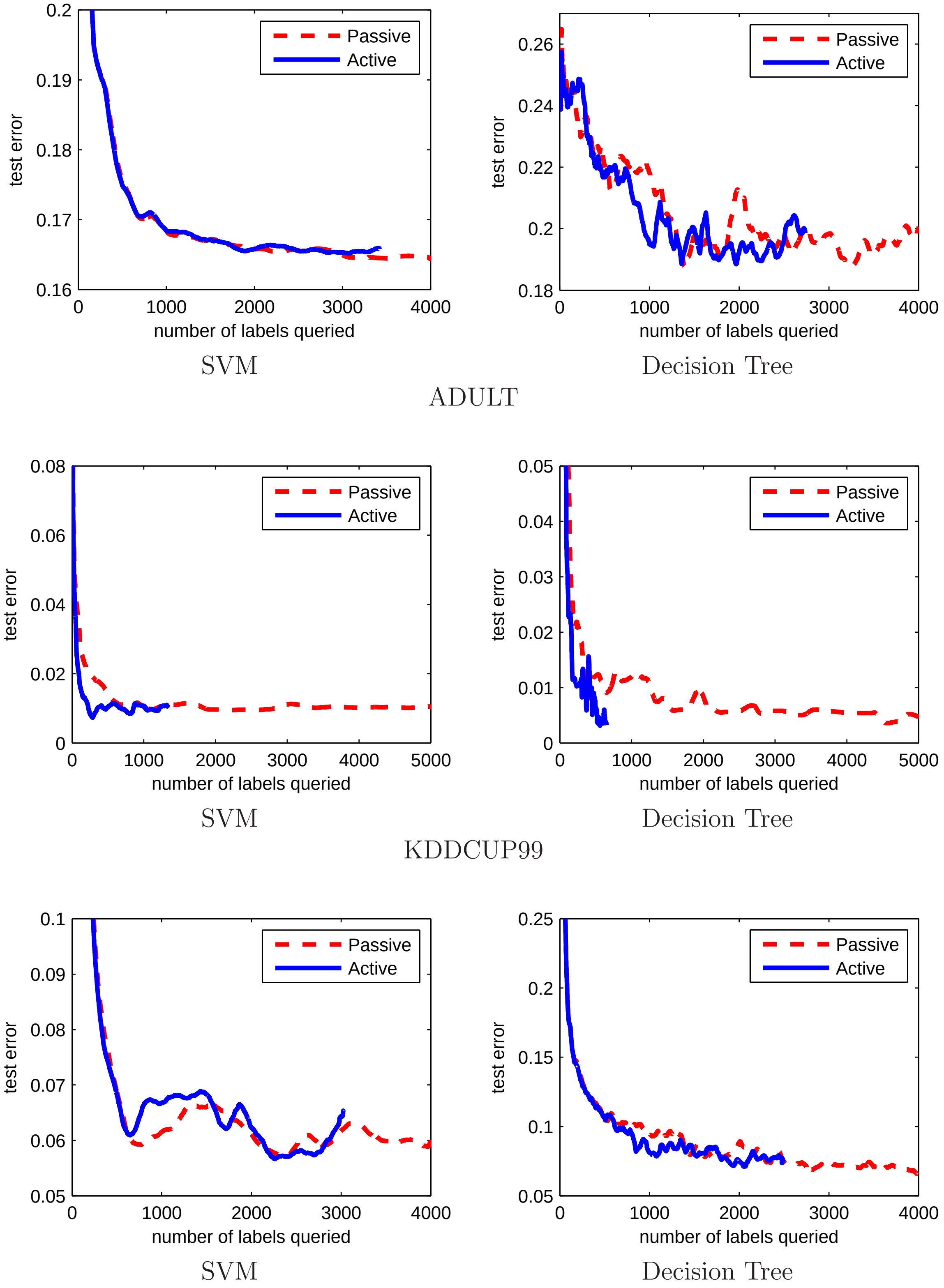
SVM Decision Tree

KDDCUP99

SVM Decision Tree

MNIST3v5

Figure 6.1: Unlabeled error rates for the binary classification experiments.



SVM Decision Tree

ADULT

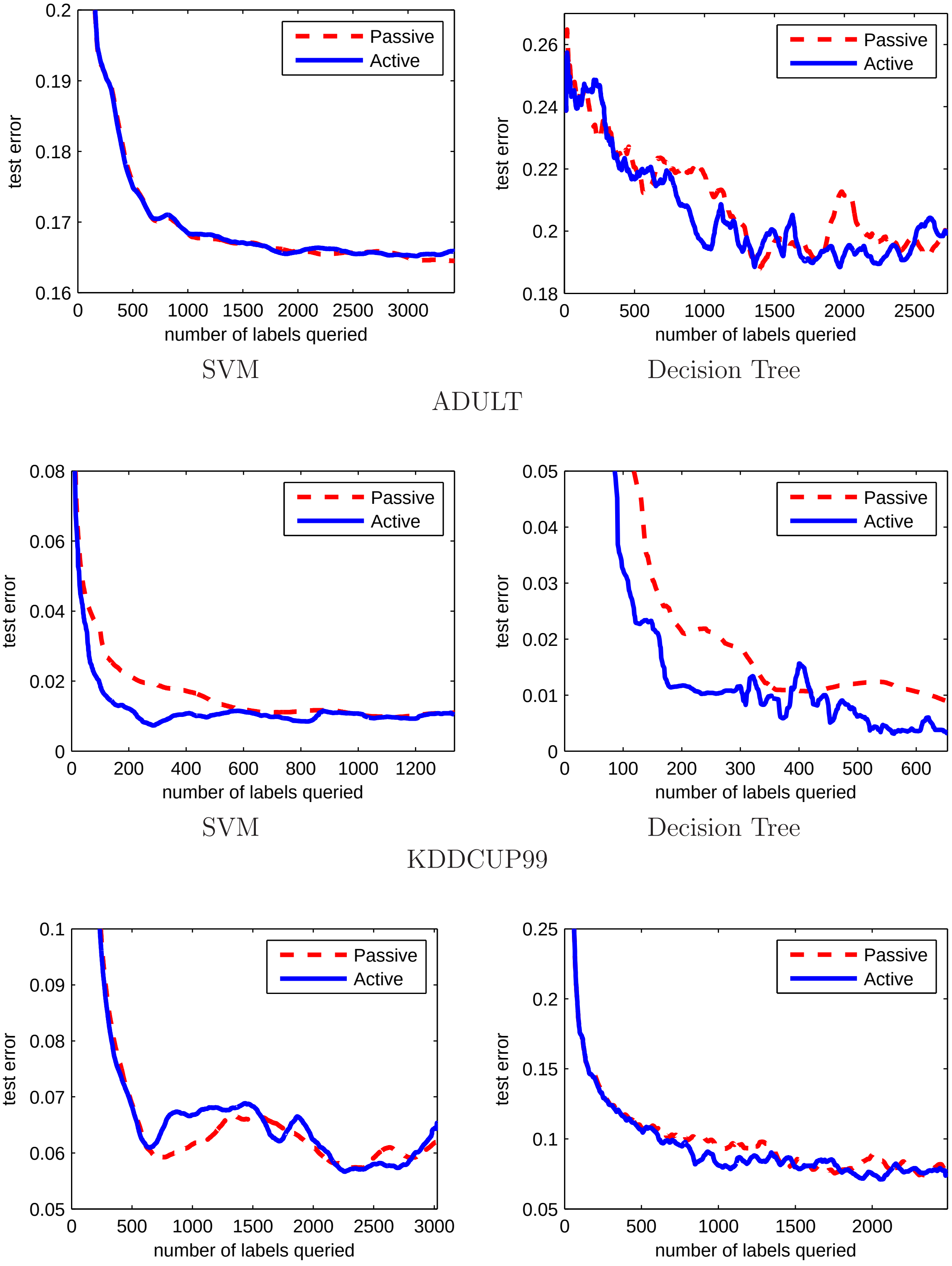
SVM Decision Tree

KDDCUP99

SVM Decision Tree

MNIST3v5

Figure 6.2: Labeled error rates for the binary classification experiments.



SVM Decision Tree

ADULT

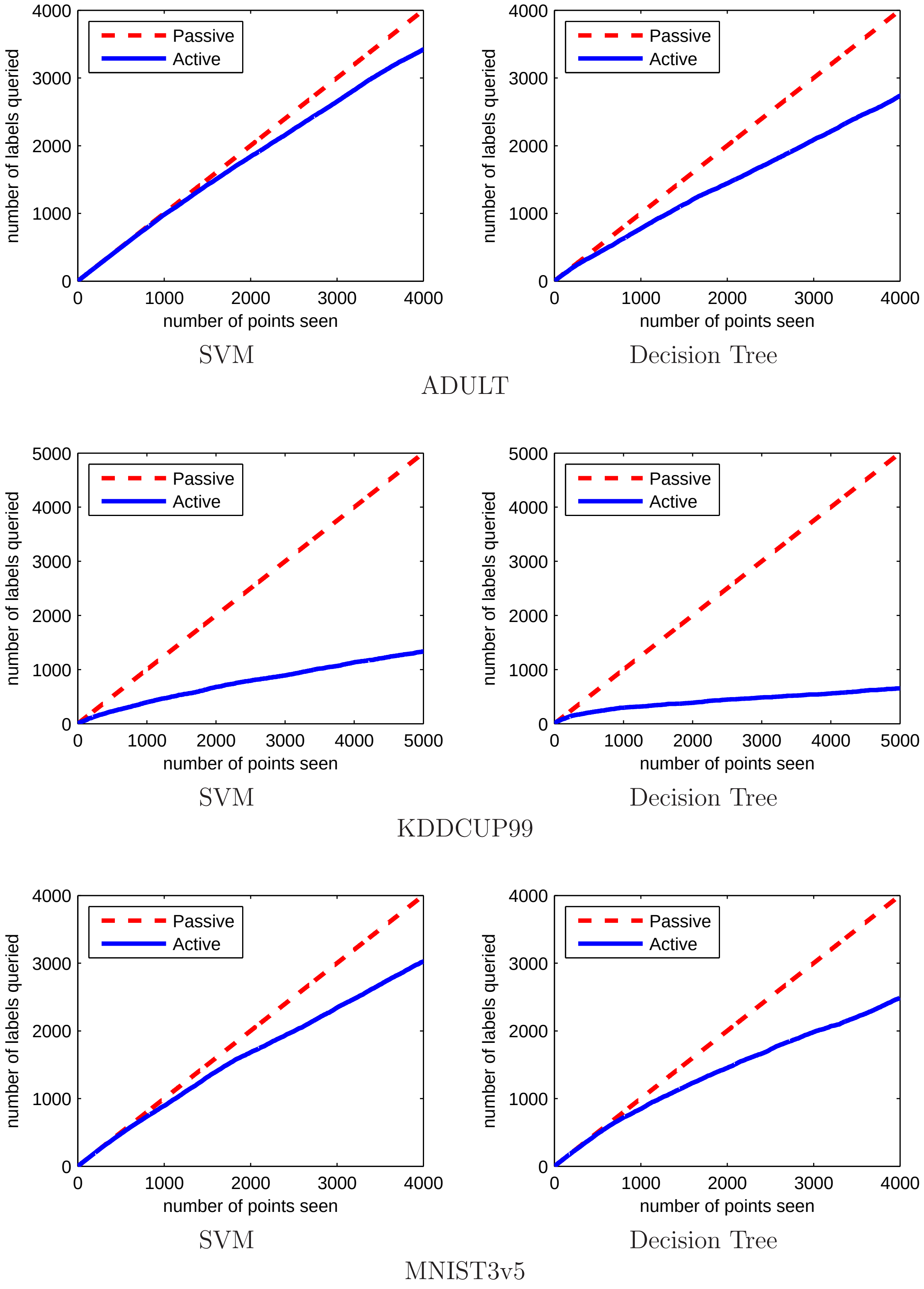
SVM Decision Tree

KDDCUP99

SVM Decision Tree

MNIST3v5

Figure 6.3: Labeled error rates (close-ups) for the binary classification experiments.



SVM

Decision Tree

ADULT

SVM

Decision Tree

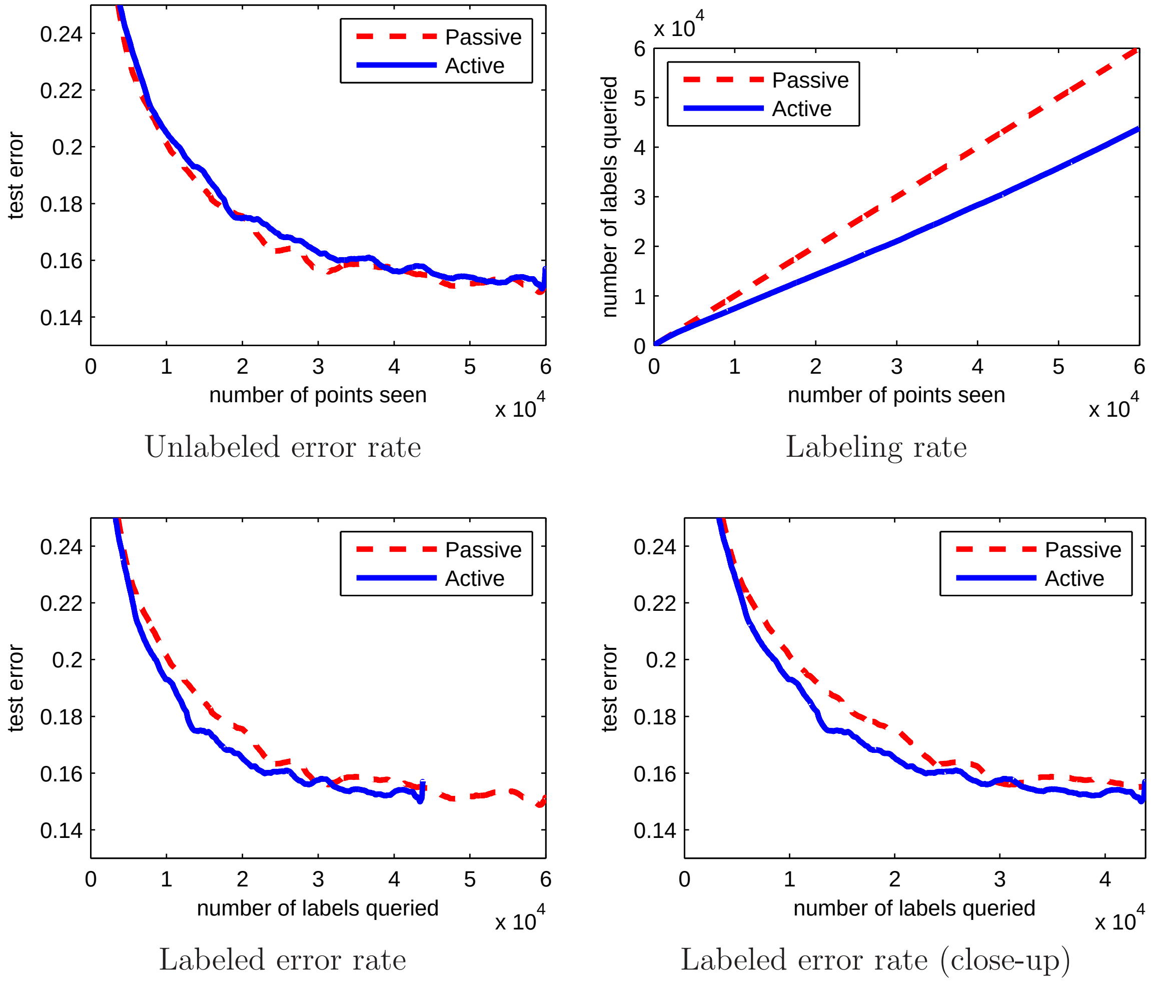
KDDCUP99

SVM

Decision Tree

MNIST3v5

Figure 6.4: Labeling rates for the binary classification experiments.



Unlabeled error rate

Labeling rate

Labeled error rate

Labeled error rate (close-up)

Figure 6.5: Error rates and labeling rates for the multi-class experiment.

Appendix A

Deviation Bounds

A.1 Finite Families of Functions

The following lemma summarizes basic Chernoff bounds estimating the bias of a coin.

Lemma A.1. *Pick any $n \geq 1$, $\eta \in (0,1)$, and function $f : \mathcal{Z} \to \{0,1\}$. Let*

$$\varepsilon_n := \frac{\log(2/\eta)}{n}.$$

Let $Z_1, \ldots, Z_n$ be n iid copies of a random variable $Z \in \mathcal{Z}$, and define

$$\mu_n(f) := \frac{1}{n} \sum_{i=1}^{n} f(Z_i).$$

With probability at least $1 - \eta$,

$$|\mu_n(f) - \mathbb{E}[f(Z)]| \leq \sqrt{\varepsilon_n/2}.$$

Also, with probability at least $1 - \eta$,

$$\mu_n(f) - \mathbb{E}[f(Z)] \leq \min\left(\sqrt{3\mathbb{E}[f(Z)] \cdot \varepsilon_n},\ 4\varepsilon_n + 2\sqrt{\mu_n(f) \cdot \varepsilon_n}\right)$$

and

$$\mathbb{E}[f(Z)] - \mu_n(f) \leq \min\left(\sqrt{2\mathbb{E}[f(Z)] \cdot \varepsilon_n},\ 2\varepsilon_n + \sqrt{2\mu_n(f) \cdot \varepsilon_n}\right).$$

To get a uniform bound for a finite family $\mathcal{F}$ of functions $f : \mathcal{Z} \to \{0,1\}$, we can simply apply a union bound.

A.2 Infinite Families of Functions

The following lemma gives a uniform bound for an infinite family of functions with finite VC dimension.

Lemma A.2 ([VC71])**.** *Pick any $n \geq 1$, $\eta \in (0,1)$, and family $\mathcal{F}$ of functions $f : \mathcal{Z} \to \{0,1\}$ with finite VC dimension. Let*

$$\varepsilon_n := \frac{4}{n} \cdot \left(\ln \mathcal{S}(\mathcal{F}, 2n) + \ln \frac{8}{\eta} \right).$$

Let $Z_1, \ldots, Z_n$ be n iid copies of the random variable $Z \in \mathcal{Z}$, and define

$$\mu_n(f) := \frac{1}{n} \sum_{i=1}^{n} f(Z_i).$$

With probability at least $1 - \eta$,

$$\mu_n(f) - \mathbb{E}[f(Z)] \leq \min\left(\varepsilon_n + \sqrt{\mathbb{E}[f(Z)] \cdot \varepsilon_n},\ \sqrt{\mu_n(f) \cdot \varepsilon_n} \right)$$

and

$$\mathbb{E}[f(Z)] - \mu_n(f) \leq \min\left(\sqrt{\mathbb{E}[f(Z)] \cdot \varepsilon_n},\ \varepsilon_n + \sqrt{\mu_n(f) \cdot \varepsilon_n} \right)$$

for all $f \in \mathcal{F}$.

Bibliography

[Ale87] K. S. Alexander. Rates of growth and sample moduli for weighted empirical processes indexed by sets. *Probability Theory and Related Fields*, 75:379–423, 1987.

[AN07] A. Asuncion and D. J. Newman. UCI machine learning repository, 2007.

[Ang98] D. Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1998.

[Ang04] D. Angluin. Queries revisited. *Theoretical Computer Science*, 313(2):175–194, 2004.

[BBL06] M.-F. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. In *Twenty-Third International Conference on Machine Learning*, 2006.

[BBZ07] M.-F. Balcan, A. Broder, and T. Zhang. Margin based active learning. In *Twentieth Annual Conference on Learning Theory*, 2007.

[BDL09] A. Beygelzimer, S. Dasgupta, and J. Langford. Importance weighted active learning. In *Twenty-Sixth International Conference on Machine Learning*, 2009.

[BEHW89] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36:929–965, 1989.

[BHW08] M.-F. Balcan, S. Hanneke, and J. Wortman. The true sample complexity of active learning. In *Twenty-First Annual Conference on Learning Theory*, 2008.

[CAL94] D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

[CN06] R. Castro and R. Nowak. Upper and lower bounds for active learning. In *Allerton Conference on Communication, Control and Computing*, 2006.

[CN07] R. Castro and R. Nowak. Minimax bounds for active learning. In *Twentieth Annual Conference on Learning Theory*, 2007.

[CSZ06] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

[CV95] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

[Das04] S. Dasgupta. Analysis of a greedy active learning strategy. In *Advances in Neural Information Processing Systems 17*, 2004.

[Das05] S. Dasgupta. Coarse sample complexity bounds for active learning. In *Advances in Neural Information Processing Systems 18*, 2005.

[DH08] S. Dasgupta and D. Hsu. Hierarchical sampling for active learning. In *Twenty-Fifth International Conference on Machine Learning*, 2008.

[DHM07] S. Dasgupta, D. Hsu, and C. Monteleoni. A general agnostic active learning algorithm. In *Advances in Neural Information Processing Systems 20*, 2007.

[DKM05] S. Dasgupta, A. Kalai, and C. Monteleoni. Analysis of perceptron-based active learning. In *Eighteenth Annual Conference on Learning Theory*, 2005.

[Fel06] V. Feldman. Optimal hardness results for maximizing agreements with monomials. In *Twenty-first Annual IEEE Computational Complexity Conference*, 2006.

[Fri09] E. Friedman. Active learning for smooth problems. In *Twenty-Second Annual Conference on Learning Theory*, 2009.

[FSST97] Y. Freund, H. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2):133–168, 1997.

[GBNT05] R. Gilad-Bachrach, A. Navot, and N. Tishby. Query by committeee made real. In *Advances in Neural Information Processing Systems 18*, 2005.

[GK06] E. Ginè and V. Koltchinskii. Concentration inequalities and asymptotic results for ratio type empirical processes. *Annals of Probability*, 34(3):1143–1216, 2006.

[GR06] V. Guruswami and P. Raghavendra. Hardness of learning halfspaces with noise. In *Forty-Seventh Annual IEEE Symposium on Foundations of Computer Science*, 2006.

[Han07] S. Hanneke. A bound on the label complexity of agnostic active learning. In *Twenty-Fourth International Conference on Machine Learning*, 2007.

[Han09] S. Hanneke. Adaptive rates of convergence in active learning. In *Twenty-Second Annual Conference on Learning Theory*, 2009.

[Hau95] D. Haussler. Sphere packing numbers for subsets of the Boolean n-cube with bounded Vapnik-Chervonenkis dimension. *Journal of Combinatorial Theory Series A*, 69(2):217–232, 1995.

[HFH$^+$09] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: An update. *SIGKDD Explorations*, 11(1):10–18, 2009.

[Kää06] M. Kääriäinen. Active learning in the non-realizable case. In *Seventeenth International Conference on Algorithmic Learning Theory*, 2006.

[KMT93] S. R. Kulkarni, S. K. Mitter, and J. N. Tsitsiklis. Active learning using arbitrary binary valued queries. *Machine Learning*, 11(1):23–35, 1993.

[Kol09] V. Koltchinskii. Rademacher complexities and bounding the excess risk in active learning. Manuscript, 2009.

[KSS94] M. Kearns, R. Schapire, and L. Sellie. Towards efficient agnostic learning. *Machine Learning*, 17:115–141, 1994.

[LB92] K. Lang and E. Baum. Query learning can work poorly when a human oracle is used. In *IEEE International Joint Conference on Neural Networks*, 1992.

[LBBH98] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[LC94] D. Lewis and J. Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Eleventh International Conference on Machine Learning*, 1994.

[LG94] D. Lewis and W. Gale. A sequential algorithm for training text classifiers. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, 1994.

[MR95] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[Qui93] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[Sau72] N. Sauer. On the density of families of sets. *Journal of Combinatorial Theory Series A*, 13:145–147, 1972.

[SC00] G. Schohn and D. Cohn. Less is more: active learning with support vector machines. In *Seventeenth International Conference on Machine Learning*, 2000.

[Set09] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[SFBL98] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *Annals of Statistics*, 26(5):1651–1686, 1998.

[SOS92] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Fifth Annual ACM Conference on Computational Learning Theory*, 1992.

[SVP06] H. Schutze, E. Velipasaoglu, and J. Pedersen. Performance thresholding in practical text classification. In *ACM International Conference on Information and Knowledge Management*, 2006.

[Tal94] M. Talagrand. Sharper bounds for Gaussian and empirical processes. *Annals of Probability*, 22(1):28–76, 1994.

[TK00] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. In *Seventeenth International Conference on Machine Learning*, 2000.

[Tsy04] A. B. Tsybakov. Optimal aggregation of classifiers in statistical learning. *Annals of Statistics*, 32(1):135–166, 2004.

[Val84] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[VC71] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and Its Applications*, 16(2):264–280, 1971.

[Wan09] L. Wang. Sufficient conditions for agnostic active learnable. In *Advances in Neural Information Processing Systems 22*, 2009.

[Zha02] T. Zhang. On the dual formulation of regularized linear systems with convex risks. *Machine Learning*, 46:91–129, 2002.

[Zha05] T. Zhang. Data dependent concentration bounds for sequential prediction algorithms. In *Eighteenth Annual Conference on Learning Theory*, 2005.