Cost Complexity of Proactive Learning via a Reduction to Realizable Active Learning

Liu Yang Jaime Carbonell

November 2009
CMU-ML-09-113

MACHINE LEARNING
DEPARTMENT

Carnegie Mellon

Cost Complexity of Proactive Learning via a Reduction to Realizable Active Learning

Liu Yang **Jaime Carbonell**

November 2009
CMU-ML-09-113

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Proactive Learning is a generalized form of active learning with multiple oracles exhibiting different reliabilities (label noise) and costs. We propose a general approach for Proactive Learning that explicitly addresses the cost vs. reliability tradeoff for oracle and instance selection. We formulate the problem in the PAC learning framework with bounded noise, and transform it into realizable active learning via a reduction technique, while keeping the overall query cost small. We propose two types of sequential hypothesis tests (denoted as **SeqHT**) that estimate the label of a given query from the noisy replies of different oracles with varying reliabilities and costs. We prove correctness and derive cost complexity of the proposed algorithms.

This research is supported by grants from the National Science Foundation.

Keywords: Proactive Learning, Learning Theory, Complexity Analysis

1 Introduction

1.1 Proactive Learning Means Options

Active learning concentrates its labeling efforts on instances that are maximally informative, due to the expense of labeling. However, it assumes a single oracle that charges uniform cost in label elicitation. In many applications, there are multiple labeling sources (a.k.a. "oracles") with different lableling accuracies and non-uniform costs; different oracles may charge differently. Hence, in order to achieve certain level of accuracy, it is not the sample complexity but the cost complexity that we ultimately care about.

Active learning does not consider the cost vs. reliability tradeoff. Cost [1] is usually regulated by the amount of work required in order to formulate an answer of certain level of accuracy. The more reliable labelers tend to charge more. The cheaper labelers however are often more noisy. Cost vs. reliability tradeoffs exist in many application domains where multiple oracles have different but unknown reliabilities. It is desirable to have cost v.s. reliability options. Consider document annotation as an example. Mechanical Turk is a cheap option; however, other than some distributional information, there is no *a priori* information on reliability of individual labelers. We may want to pay a higher price to hire linguists for reliable annotation if these really matter, or perhaps pay a medium price to hire linguistic students. In zero-sum game theory (e.g. chess) the proability of accurate determination of an optimal move (the "label" given the game state) is often a function of look-ahead depth, which itself is an exponential function of computational cost incurred. It is also a function of the accuracy of the state-evaluation method, hence different game players will have different accuracies, related to but not determined by cost.

In astrophysics, if we want to learn a model for galactic formation and evolution, complex simulations to predict outcomes given initial conditions accurately can take weeks or months of supercomputer time, whereas coarser models can yield less relibale answers much faster (in minutes or hours). Simulation-model complexity causes a cost-accuracy tradeoff in many scientific computing application, and a proactive learning needs to select the best model as well as most informative instance to obtain prediction (labels) to train the learner.

First described in [Donmez & Carbonell, 2008, Donmez et al., 2009], Proactive Learning also considers the case of reluctant oracles, which may fail to answer. In this paper does not consider reluctance but note it can be reduced to accuracy loss by simply predicting the majority class upon oracle failure, or reduced to increased cost by trying other oracles until one yields an answer. [Donmez & Carbonell, 2008] formulate Proactive Learning as a utility optimization problem given a budget. [Donmez et al., 2009] estimates a confidence interval for expert reliability.

Proactive Learning introduces oracle selection contitioned on a cost vs. estimated reliability (or noise rate) tradeoff. Active learning assumes that if the oracle is noisy or expensive, it is has no other choice. Moreover, Proactive Learning can query multiple oracles under the bounded-noise assumption to increase reliability of the ensemble answer by incurring extra cost. In essence, it decides *where* to get the labling information as well as *what* instances to query, and is therefore more flexible. The approach we propose in this paper captures the tradeoff and extra degree of

[1] The cost of labeling is different from the cost of misclassification in cost-sensitive learning.

freedom by clearly formulating them into the learning model.

The central task of Proactive Learning is to reduce cost with a certain reliability guarantee. The cost can be time, computation, human effort, or real money. Proactive learning procedures minimize the total cost instead of the number of queries, accounting for oracle cost differentials. Hence, we are interested in cost complexity instead of sample complexity.

This paper is the first theoretical work on Proactive Learning. We give a mathematical definition of Proactive Learning and model it in the PAC learning framework under bounded noise. We choose to focus on oracle selection – the aspect that is different from active learning. We generate a noisy answer ensemble from which we estimate the best label when multiple oracles are queried. We propose two types of SeqHT [2] that make the cost vs. reliability tradeoff explicit. We derive cost complexity bounds for each of the proposed proactive learning algorithms. For the task of deciding which points to query, we simply reduce to the instance selection teqchniques from standard active learning.

We choose to study bounded noise for proactive learning for the following reasons: Uniform noise is too simplistic and unrealistic since not all instances are equally simple to label. Bounded noise [Angluin & Larid, 1987] assumes stochastic labeling from a given oracle for a sequence of examples: noise rates for different examples can be different but are always bounded away from $1/2$. The agnostic setting is a more severe situation in which label noise can exceed $1/2$ and possibly the optimal classifier $f^* \notin \mathcal{F}$. Although progress has been made in agnostic active learning [Balcan et al., 2006], given the additional complexities of proactive learning, it becomes more difficult to formulate an objective with an arbitrary noise distribution.

In order to have the concept of "optimal" oracle selection, we need to address how oracles are related to ground truth, and specifically to the target function. In the bounded noise situation, there exists $f^* \in \mathcal{F}$ with which all oracles agree. Given the property that label noise is bounded away from $1/2$, all oracles more often than not agree with f^*. And the more oracles queried the higher the probability of estimating ground truth. This is the advantage of formulating proactive learning under bounded noise. Finally, we leave out malicious noise [Laird, 1988] out of this study, as it as a rare type of noise in practice under which the oracle may corrupt not only the labels but the underlying distribution of examples: it makes the decision on whether or not to flip the label of a given example, dependent on its features.

The reminder of the paper has the following structure: In Section 2, we give a formal definition of Proactive Learning, and propose a learning framework ProAL via a reduction to realizable active learning. We then introduce n-Threaded SeqHTs in Section 3 and central pool based SeqHT in Section 4, as two instantiations of SeqHTRoutine in ProAL, with analyses of their cost complexity. In Section 5, we propose a procedure that aggregates n-Threaded SeqHTs and the central pool based SeqHT. In Section 6, we discuss the online nature of oracle reliability estimation and future work stemming therefrom.

[2] In this work, we assume the oracles are non-persistent : they may give different answers for the same example at different time.

2 Formal Definition of Proactive Learning

Consider the binary classification task. Let $\mathcal{F}$ be the hypothesis class. Let $\mathcal{X}$ represent the data space, and $\mathcal{Y} \in \pm 1$ represent the label space. There are n oracles available for querying. Instead of a single oracle, we consider the active learning problem with n noisy oracles. For the jth oracle, its cost is c_j and its unknown noise rate bound is $\alpha_j \geq 0$, and $j = 1, \cdots, n$.

There is a special $f^* \in \mathcal{F}$, called the target function. The goal of Proactive learner is, given the noisy answers from n oracles, to output a classifier f whose generalization error $\mathbb{P}(f(\mathbf{x}) \neq f^*(\mathbf{x})) \leq \epsilon$ where $\mathbf{x} \in \mathcal{X}$, with high probability $1 - \delta$, while keeping the total query cost small (Note f and f^* are same for all oracles). To this end, we assume oracle noise to be the bounded rate class noise [Angluin & Larid, 1987], meaning $\mathbb{P}(y \neq f^*(\mathbf{x})|\mathbf{x}) \leq \alpha_j$ where the probability is over y from oracle j. The noise rate is bounded away from $1/2$; thus $0 \leq \alpha_j < 1/2$, for $j = 1, \cdots, n$.

We propose a meta-procedure ProAL that takes a realizable active learning algorithm **A**. The realizable algorithm **A** has some stopping criterion built into it: it halts either because it has made enough queries, or because it meets a budge on the total cost. **A** has the following property : it halts and outputs a classifier after making $N(\epsilon, \delta)$ label requests; and with probability $1 - \delta/2$, the output classifier has error rate at most ϵ. The realizable sample complexity of active learning $N(\epsilon, \delta)$ has been bounded in a variety of scenarios in [Dasgupta, 2005, Hanneke, 2007, Hanneke, 2009]. ProAL lets **A** choose examples to query, and hands back to **A** the "true" label after calling SeqHTRoutine : it runs variants of SeqHT on the noisy answers returned by n oracles.

Proactive Learner (Input a realizable algorithm **A**) (denoted as ProAL)
Initialize $i = 0$
1. do
2. $\quad i = i + 1$
3. $\quad$ Let A choose a query point $\mathbf{x}$ from unlabeled data
4. $\quad$ Let y =SeqHTRoutine($\mathbf{x}$, $\delta/(4i^2)$)
5. $\quad$ Return y into **A**
6. until **A** halts

We split δ into two equal parts, covering the two ways in which ProAL may fail. First, ProAL may fail because **A** fails in the realizable case with probability $\delta/2$. **A** succeeds if it outputs a classifier whose error rate at most ϵ. The corresponding sample complexity of **A** is $N(\epsilon, \delta/2)$. We cover this by setting the accuracy parameter and confidence parameter of **A** to $(\epsilon, \delta/2)$. Second, ProAL may fail because one or more of its calls of SeqHTRoutine fail. In case it does not fail, the number of calls of SeqHTRoutine is upper bounded by $N(\epsilon, \delta/2)$. Putting together the two parts, the total probability of either of the two failures occurring is at most δ by union bound. For the second half, we further split the remaining $\delta/2$ to the calls of SeqHTRoutine and choose the condence parameter adaptively:

$$\delta'_i = \delta/(4i^2) \tag{1}$$

for ith call SeqHTRoutine, with $i = 1, \cdots, N(\epsilon, \delta/2)$. The adaptive condence parameters satisfy

$\sum_{i=1}^{N(\epsilon,\delta/2)} \delta/(4i^2) < \sum_{i=1}^{\infty} \delta/(4i^2) = \delta/2$ followed from the property that

$$\sum_{i=1}^{\infty} 1/(2i^2) = \pi^2/12 < 1 \tag{2}$$

By union bound, the calls of SeqHTRoutine give the correct answer with high probability $1 - \delta/2$. In Section 3 and Section 4, we propose two types of SeqHTRoutine for Proactive Learner and analyze their cost complexities.

3 n-Threaded Proactive Learner

SequentialTest1($\mathbf{x}$, δ') (denoted as ST1)
Initialize the set of labels $z_j = \emptyset$ for $j = 1, \cdots, n$
do
  Select Oracle j that minimizes $(|z_j| + 1)c_j$
  Query Oracle j and get label l
  $z_j \leftarrow z_j \cup \{l\}$ (z_j is a multi-set of ± 1s)
  $p_j \leftarrow$ average of elements in z_j

$$I_j \leftarrow \left[p_j - \sqrt{\frac{2\ln(4n|z_j|^2/\delta')}{|z_j|}}, p_j + \sqrt{\frac{2\ln(4n|z_j|^2/\delta')}{|z_j|}} \right]$$

while $0 \in I_j$
if $I_j \subset (-\infty, 0)$, output -1
else output $+1$

For a given example $\mathbf{x}$, we apply an adaptive sampler (ST1) to find out its "true" label based on oracles' noisy answers, while keeping the overall cost small. The sample size is a random variable dependent on samples already seen.

To keep expense on each oracle equal, we dynamically choose oracles so that we do not waste too much on the wrong ones. ST1 chooses the oracle that increases the maximum spent by the smallest amount. Sometimes if an oracle has very large cost, we do not pick it even it has spent the least so far. If n oracles all have equal costs, ST1 uniformly spreads around on which oracle to query.

Each oracle runs a separate SeqHT based on the sample it gets. Whoever among the n-threaded SeqHTs halts first will return the correct label for $\mathbf{x}$. Using ST1 as SeqHTRoutine in ProAL, we have our first algorithm (denoted as tProAL). The correctness of the returned label, is guaranteed by the correctness of SeqHT [Wald, 1945]. Although SeqHT has been used for active learning [Kääriäinen, 2006, Bar-Yossef, 2003, Hanneke, 2007], no explicit results on cost complexity have been derived for Proactive learning.

Note for each oracle, SeqHT is the optimal test on a given a dataset. Our task is essentially related to Likelihood Ratio Test that achieves the optimal rate [Wald, 1945]. $1/\sqrt{n}$ is proved to be the optimal rate for SeqHT on a Bernoulli parameter p up to a constant factor, i.e. one can not improve over SeqHT by running a different test. The Hoeffding inequality offers the best test. No significant improvement can be obtained by replacing Hoeffding inequality with any other inequality to change the way of calculating confidence interval. If the only change made to the test is the way of calculating confidence interval, one can at most save a $\log\left(\frac{1}{1-2\alpha}\right)$ factor, where α is the noise rate for a given oracle.

Theorem 1. $\exists k \in (0,\infty)$ *such that the cost complexity of tProAL is upper bounded by*

$$\min_{j=1,\cdots,n} \frac{knc_j}{(1/2-\alpha_j)^2} \ln\left(\frac{nN(\epsilon,\delta/2)}{\delta(1/2-\alpha_j)}\right) N(\epsilon,\delta/2) \tag{3}$$

where $N(\epsilon,\delta/2)$ *is the sample complexity for the realizable active learning algorithm.*

Proof. We split the failure probability δ' in calling of ST1, equally into n parts : each part corresponds to a SqeHT for one individual oracle. For arbitrary oracle j, we further split δ'/n into $|z_j|$ parts. Note we do not know how many queries SeqHT j exactly processes till ST1. Denote M_j as $|z_j|$ when ST1 halts and p_j as SeqHT sample mean on oracle j.

We first prove the correctness of tProAL. Define

$$\epsilon_{j,|z_j|} \;=\; \sqrt{2\ln\left(\frac{4n|z_j|^2}{\delta'}\right)/|z_j|} \tag{4}$$

By Hoeffding inequality and the union bound

$$\begin{aligned}
&\mathbb{P}(\text{No SeqHT of any of the } n \text{ oracles ever halt})\\
\leq\;& \sum_{j=1}^{n}\sum_{|z_j|}\mathbb{P}(|p_j - Ep_j| > \epsilon_{j,|z_j|})\\
\leq\;& \sum_{j=1}^{n}\sum_{|z_j|} 2\exp(-|z_j|\epsilon_{j,|z_j|}^2/2) \;\leq\; \delta'
\end{aligned}$$

Equivalently, with probability $\geq 1-\delta'$, we have the following invariant based on definition (4) of $\epsilon_{j,|z_j|}$:
At all time in ST1, $\exists j$, s.t.

$$|p_j - Ep_j| \;\leq\; \epsilon_{j,|z_j|}$$

We now show under the definition (4) of $\epsilon_{j,|z_j|}$, the following inequality holds :

$$\sum_{j=1}^{n}\sum_{|z_j|} 2\exp(-|z_j|\epsilon_{j,|z_j|}^2/2) \;\leq\; \delta' \tag{5}$$

Sufficient if $\forall j, \forall |z_j|$,

$$\exp(-|z_j|\epsilon_{j,|z_j|}^2/2) \;\leq\; \frac{\delta'}{4n|z_j|^2} \tag{6}$$

From Property (2), it follows that

$$\sum_{|z_j|=1}^{\infty} \frac{\delta'/n}{4|z_j|^2} \quad \leq \quad \frac{\delta'/n}{2} \tag{7}$$

If Inequality (6) holds, combined with Inequality (7), we get

$$\sum_{j=1}^{n} \sum_{|z_j|} 2\exp(-|z_j|\epsilon_{j,|z_j|}^2/2) \leq \sum_{i=1}^{n} \delta'/n \leq \delta'$$

Thus Inequality (5) holds. Inequality (6) is true under Definition (4) and Inequality (6) is proved a sufficient condition for Inequality (5). Therefore Inequality (5) holds with Definition (4). The confidence interval in Step 6 of ST1 is defined using exactly the $\epsilon_{j,|z_j|}$ defined in (4), thus tProAL is correct.

We then compute the cost complexity of ST1. By the stopping criterion of ST1, at any moment in running ST1, $|p_j| \leq \epsilon_{j,|z_j|}$ has to be true (otherwise it halts). $|p_j - Ep_j| \leq \epsilon_{j,|z_j|}$ is always true for ST1 . By triangle inequality, the invariant is

$$|Ep_j| \leq |p_j| + |Ep_j - p_j| \leq 2\epsilon_{j,|z_j|} \tag{8}$$

The absolute value of expectation of p_j is bounded as

$$|Ep_j| \geq |-\alpha_j + (1-\alpha_j)| = 1 - 2\alpha_j \tag{9}$$

as $0 \leq \alpha < 1/2$. Thus

$$1 - 2\alpha_j \quad \leq \quad 2\epsilon_{j,|z_j|} \tag{10}$$

Plug Definition (4) of $\epsilon_{j,|z_j|}$ into Inequality (10), we get

$$1 - 2\alpha_j \leq 2\sqrt{2\ln\left(\frac{4nM_j^2}{\delta'}\right)/M_j}$$

Therefore, we get an upper bound on M_j when ST1 halts

$$M_j \quad \leq \quad \frac{16}{(1-2\alpha_j)^2} \ln\left(\sqrt{\frac{4n}{\delta'}} M_j\right) \tag{11}$$

Inequality (11) can be simplified by using the following property [Vidyasagar, 2003]: for $u > 0$, $v > 0$, $w > 0$ and $uv\mathbf{e}^{w/u} > 4\log_2 \mathbf{e}$,

$$m \leq w + u\ln(vm) \quad \Rightarrow \quad m \leq 2w + 2u\ln(uv) \tag{12}$$

from which we get

$$M_j \quad \leq \quad \frac{32}{(1/2-\alpha_j)^2} \ln\left(\sqrt{\frac{4n}{\delta'}} \frac{16}{(1/2-\alpha_j)}\right) \tag{13}$$

Equation (1) combined with $i \le N(\epsilon, \delta/2)$ implies

$$1/\delta' \le \frac{4N^2(\epsilon,\delta/2)}{\delta} \tag{14}$$

Combining the upper bound (14) on $1/\delta'$ with Inequality (13), we get

$$\begin{aligned} M_j &\le \frac{32}{(1/2-\alpha_j)^2}\ln\left(\sqrt{\frac{n}{\delta}}\frac{64N(\epsilon,\delta/2)}{(1/2-\alpha_j)}\right) \\ &\le \frac{32}{(1/2-\alpha_j)^2}\ln\left(\frac{64nN(\epsilon,\delta/2)}{\delta(1/2-\alpha_j)}\right) \end{aligned} \tag{15}$$

Notice that $\sqrt{n/\delta} \le n/\delta$, as $\sqrt{n/\delta} \ge 1$. Denote

$$\begin{aligned} \text{MinCost} &= \min_{j=1,\cdots,n} M_j c_j \\ &\le \min_{j=1,\cdots,n} \frac{32c_j}{(1/2-\alpha_j)^2}\ln\left(\frac{64nN(\epsilon,\delta/2)}{\delta(1/2-\alpha_j)}\right) \end{aligned} \tag{16}$$

from inequality (15). Let oracle j^* be the minimizer in MinCost. It immediately follows that

$$M_{j^*}c_{j^*} \le \text{MinCost}$$

Our oracle selection strategy maintains the following invariant (17) at any moment of ST1. We prove this by contradiction. Consider the round at which Inequality (17) is violated: suppose oracle j is selected; $|z_j|$ increases by 1 and violates Inequality (17). Then there must exists another oracle j' with $c_{j'}(|z_{j'}|+1) \le c_j(|z_j|+1)$. This contradicts to our oracle selection strategy: choose the oracle that increases the maximum spent by the smallest amount. Thus Inequality (17) has to be true at any moment of ST1 .

$$\begin{aligned} \max_{j'=1,\cdots,n} c_{j'}|z_{j'}| &\le \min_{j'=1,\cdots,n} c_{j'}(|z_{j'}|+1) \\ &\le c_{j^*} + \text{MinCost} \\ &\le 2\text{MinCost} \end{aligned} \tag{17}$$

From Invariant (17), we conclude that the cost complexity of ST1 is upper bounded by $2n$MinCost. The realizable algorithm **A** hands to ST1 at most $N(\epsilon, \delta/2)$ examples to query their labels. Thus the cost complexity of tProAL

$$\begin{aligned} &\le 2n\text{MinCost}N(\epsilon,\delta/2) \\ &\le \min_{j=1,\cdots,n} \frac{64nc_j}{(1/2-\alpha_j)^2}\ln\left(\frac{64nN(\epsilon,\delta/2)}{\delta(1/2-\alpha_j)}\right)N(\epsilon,\delta/2) \end{aligned}$$

by combining Inequality (16). □

Note that even if we know which oracle to use ahead of time (without going through the threaded SeqHT), it only saves a factor of $2n$ and nothing more than that, compared to which, the cost complexity bound achieved by tProAL is not too much worse.

4 Central Pool based Proactive Learner

SequentialTest2(x, δ') (denoted as ST2)
Initialize the set of labels $z_j = \emptyset$ for $j = 1, \cdots, n$
1. **do**
2. Select Oracle j that minimizes $(|z_j| + 1)c_j$
3. Query Oracle j and get label l
4. $z_j \leftarrow z_j \cup \{l\}$
5. $p \leftarrow$ average of elements in $\mathcal{Z} = \cup_{j=1}^{n} z_j$
6.

$$I \leftarrow \left[p - \sqrt{2 \ln \left(\frac{4|\mathcal{Z}|^2}{\delta'} \right) / |\mathcal{Z}|}, p + \sqrt{2 \ln \left(\frac{4|\mathcal{Z}|^2}{\delta'} \right) / |\mathcal{Z}|} \right]$$

7. **while** $0 \in I$
8. **if** $I \subset (-\infty, 0)$, output -1
9. **else** output $+1$

ST2 has the same oracle-sampling strategy with ST1. But it maintains a central pool to include samples from oracles and run a single SeqHT. It returns the correct label when the single SeqHT halts. The hope is a single SeqHT can save cost complexity than n-threaded in some scenario. Using ST2 as SeqHTRoutine in ProAL, we obtain our second algorithm (denoted as cProAL).

Theorem 2. $\exists k_1, k_2 \in (0, \infty)$ *such that the cost complexity of cProAL is upper bounded by*

$$\left(\frac{k_1 \sum_{j=1}^{n} 1/c_j}{\left(\sum_{j=1}^{n} \beta_j / c_j \right)^2} \ln \left(\frac{N(\epsilon, \delta/2)}{\delta} \frac{\sum_{j=1}^{n} 1/c_j}{\sum_{j=1}^{n} \beta_j / c_j} \right) + \frac{k_2 \sum_{j=1}^{n} \beta_j}{\sum_{j=1}^{n} \beta_j / c_j} + \frac{n}{\sum_{i=1}^{n} 1/c_j} \right) n N(\epsilon, \delta/2) \tag{18}$$

where $\beta_j = 1 - 2\alpha_j$. $N(\epsilon, \delta/2)$ *is the sample complexity for the realizable active learning algorithm.*

Proof. We split the failure probability δ' in calling of ST2 into $|\mathcal{Z}|$ parts. We do not know how many queries ST2 exactly processes till it halts. Denote M as $|\mathcal{Z}|$ when ST2 halts and p as the sample mean.

We first prove the correctness of tProAL. Define

$$\epsilon_{|\mathcal{Z}|} = \sqrt{2 \ln \left(\frac{4|\mathcal{Z}|^2}{\delta'} \right) / |\mathcal{Z}|} \tag{19}$$

By Hoeffding inequality and the union bound

$$\begin{aligned}\mathbb{P}(\text{ST2 not halting}) &\le \sum_{|\mathcal{Z}|=1,2,\cdots} \mathbb{P}(|p - Ep| > \epsilon_{|\mathcal{Z}|}) \\ &\le \sum_{|\mathcal{Z}|=1,2,\cdots} 2\exp(-|\mathcal{Z}|\epsilon_{|\mathcal{Z}|}^2/2) \\ &\le \delta'\end{aligned}$$

Equivalently, with probability $\ge 1-\delta'$, we have the following invariant based on definition (19) of $\epsilon_{|\mathcal{Z}|}$:
At all time in ST2, $\exists j$, s.t.

$$|p_j - Ep_j| \le \epsilon_{|\mathcal{Z}|}$$

We now show under the definition (19) of $\epsilon_{|\mathcal{Z}|}$, the following inequality holds :

$$\sum_{|\mathcal{Z}|=1,2,\cdots} 2\exp(-|\mathcal{Z}|\epsilon_{|\mathcal{Z}|}^2/2) \le \delta' \tag{20}$$

sufficient if $\forall|\mathcal{Z}|$,

$$\exp(-|\mathcal{Z}|\epsilon_{|\mathcal{Z}|}^2/2) \le \frac{\delta'}{4|\mathcal{Z}|^2} \tag{21}$$

From Property (2), it follows that

$$\sum_{|\mathcal{Z}|=1,2,\cdots} \frac{\delta'}{4|\mathcal{Z}|^2} \le \sum_{|\mathcal{Z}|=1}^{\infty} \frac{\delta'}{4|\mathcal{Z}|^2} \le \frac{\delta'}{2} \tag{22}$$

If Inequality (21) holds, combined with Inequality (22), we get

$$\sum_{|\mathcal{Z}|=1,2,\cdots} 2\exp(-|\mathcal{Z}|\epsilon_{\mathcal{Z}}^2/2) \le \delta'$$

Thus Inequality (20) holds. Inequality (21) is true under Definition (19), and Inequality (21) is proved a sufficient condition for Inequality (20). Therefore Inequality (20) holds with Definition (19). The confidence interval in Step 6 of ST2 is defined using exactly the $\epsilon_{|\mathcal{Z}|}$ defined in (19); thus cProAL is correct.

We then compute the cost complexity of cProAL. By the stopping criterion of ST2, at any moment in running ST2, $|p| \le \epsilon_{|\mathcal{Z}|}$ has to be true (otherwise it halts).
$|p - Ep| \le \epsilon_{|\mathcal{Z}|}$ is always true for ST2 . By triangle inequality, the invariant is

$$|Ep| \le |p| + |Ep - p| \le 2\epsilon_{|\mathcal{Z}|} \tag{23}$$

We know

$$|Ep_j| \ge |-\alpha_j + (1-\alpha_j)| = 1 - 2\alpha_j \tag{24}$$

as $0 \leq \alpha < 1/2$. Denote $\beta_j = 1 - 2\alpha_j$. From Inequality (24), we have

$$|Ep| \geq \sum_{j=1}^{n} \frac{|Ep_j||z_j|}{|\mathcal{Z}|} \geq \frac{\sum_{j=1}^{n} \beta_j |z_j|}{|\mathcal{Z}|} \tag{25}$$

Combining Inequality (25) and Inequality (23), we get

$$\frac{\sum_{j=1}^{n} \beta_j |z_j|}{|\mathcal{Z}|} \quad \leq \quad 2\epsilon_{|\mathcal{Z}|} \tag{26}$$

Plug Definition (19) of $\epsilon_{|\mathcal{Z}|}$ into Inequality (26), we get

$$\frac{\sum_{j=1}^{n} \beta_j |z_j|}{|\mathcal{Z}|} \quad \leq \quad \sqrt{8 \ln\left(\frac{4|\mathcal{Z}|^2}{\delta'}\right) / |\mathcal{Z}|} \tag{27}$$

In order to compute the cost complexity from Inequality (27), our strategy is to first derive upper and lower bounds on $|z_j|$ from the invariant of ST2. Combining with the upper bound of $|z_j|$, we get an upper bound on the cost complexity $\sum_{j=1} c_j |z_j|$, which only involves $|\mathcal{Z}|$. Then we only need to bound $|\mathcal{Z}|$ from above. This can be done by combining the lower bound on $|z_j|$ with Inequality (27), and solving for $|\mathcal{Z}|$. Below gives the details of proof.

cProAL shares the same oracle selection strategy with cProAL ; thus maintains the same invariant (17)

$$\begin{aligned} c_j |z_j| \quad &\leq \quad \max_{j'=1,\cdots,n} c_{j'} |z_{j'}| \\ &\leq \quad \min_{j'=1,\cdots,n} c_{j'} (|z_{j'}| + 1) \leq c_j \left(|z_j| + 1\right) \end{aligned} \tag{28}$$

By Invariant (28), the following holds

$$\begin{aligned} c_j \left(|z_j| + 1\right) \quad &= \quad \left(\sum_{j=1}^{n} \frac{1/c_j}{\sum_{j'=1}^{n} 1/c_{j'}}\right) c_j \left(|z_j| + 1\right) \\ &\geq \quad \sum_{j=1}^{n} \frac{1/c_j}{\sum_{j'=1}^{n} 1/c_{j'}} \left(\max_{j'=1,\cdots,n} c_{j'} |z_{j'}|\right) \\ &\geq \quad \frac{\sum_{j=1}^{n} |z_j|}{\sum_{j=1}^{n} 1/c_j} = \frac{|\mathcal{Z}|}{\sum_{j=1}^{n} 1/c_j} \end{aligned}$$

which implies a lower bound on $|z_j|$

$$|z_j| \quad \geq \quad \frac{1/c_j}{\sum_{j'=1}^{n} 1/c_{j'}} |\mathcal{Z}| - 1 \tag{29}$$

Again by Invariant (28), the following holds

$$\begin{aligned} c_j|z_j| &\leq \min_{j=1,\cdots,n} c_j(|z_j|+1) \\ &= \sum_j \frac{1/c_j}{\sum_{j'=1}^n 1/c_{j'}} \left(\min_{j'=1,\cdots,n} c_{j'}\left(|z_{j'}|+1\right)\right) \\ &\leq \frac{\sum_{j=1}^n \left(|z_j|+1\right)}{\sum_{j=1}^n 1/c_j} \\ &\leq \frac{|\mathcal{Z}|+n}{\sum_{j=1}^n 1/c_j} \end{aligned}$$

which implies an upper bound on $|z_j|$

$$|z_j| \leq \frac{1/c_j}{\sum_{j'=1}^n 1/c_{j'}} \left(|\mathcal{Z}|+n\right) \tag{30}$$

Combining the lower bound (29) on $|z_j|$ with Inequality (27), we get

$$\frac{\left(\frac{\sum_{j=1}^n \beta_j/c_j}{\sum_{j=1}^n 1/c_j}|\mathcal{Z}| - \sum_{j=1}^n \beta_j\right)^2}{|\mathcal{Z}|} \leq 16\ln\left(\sqrt{\frac{4}{\delta'}}|\mathcal{Z}|\right)$$

We expand the above inequality and get

$$\begin{aligned} &\left(\frac{\sum_{j=1}^n \beta_j/c_j}{\sum_{j=1}^n 1/c_j}\right)^2 |\mathcal{Z}| - 2\frac{\sum_{j=1}^n \beta_j/c_j \sum_{j=1}^n \beta_j}{\sum_{j=1}^n 1/c_j} \\ &\qquad \leq 16\ln\left(\sqrt{\frac{4}{\delta'}}|\mathcal{Z}|\right) - \frac{\left(\sum_{j=1}^n \beta_j\right)^2}{|\mathcal{Z}|} \end{aligned}$$

Since $\left(\sum_{j=1}^n \beta_j\right)^2/|\mathcal{Z}| \geq 0$, we have

$$|\mathcal{Z}|\left(\frac{\sum_{j=1}^n \frac{\beta_j}{c_j}}{\sum_{j=1}^n \frac{1}{c_j}}\right)^2 - \frac{2\sum_{j=1}^n \frac{\beta_j}{c_j}\sum_{j=1}^n \beta_j}{\sum_{j=1}^n 1/c_j} \leq 16\ln\left(\sqrt{\frac{4}{\delta'}}|\mathcal{Z}|\right)$$

Simplifying the above inequality, we get

$$\begin{aligned} |\mathcal{Z}| &\leq \frac{2\sum_{j=1}^n \beta_j \sum_{j=1}^n 1/c_j}{\sum_{j=1}^n \beta_j/c_j} \\ &+ 16\left(\frac{\sum_{j=1}^n 1/c_j}{\sum_{j=1}^n \beta_j/c_j}\right)^2 \ln\left(\sqrt{\frac{4}{\delta'}}|\mathcal{Z}|\right) \end{aligned}$$

Using Property (12), we get an upper bound on $|\mathcal{Z}|$

$$
\begin{aligned}
|\mathcal{Z}| \quad &\leq \quad \frac{4\sum_{j=1}^n \beta_j \sum_{j=1}^n 1/c_j}{\sum_{j=1}^n \beta_j/c_j} \\
&+ \quad 32\left(\frac{\sum_{j=1}^n 1/c_j}{\sum_{j=1}^n \beta_j/c_j}\right)^2 \ln\left(\frac{32}{\sqrt{\delta'}}\left(\frac{\sum_{j=1}^n 1/c_j}{\sum_{j=1}^n \beta_j/c_j}\right)^2\right)
\end{aligned} \tag{31}
$$

Equation (1) combined with $i \leq N(\epsilon, \delta/2)$ implies

$$
1/\delta' \quad \leq \quad \frac{4N^2(\epsilon, \delta/2)}{\delta} \tag{32}
$$

Combining the upper bound (32) on $1/\delta'$ with Inequality (31), we get an upper bound on $|\mathcal{Z}|$

$$
\begin{aligned}
|\mathcal{Z}| \leq \frac{32\left(\sum_{j=1}^n 1/c_j\right)^2}{\left(\sum_{j=1}^n \beta_j/c_j\right)^2} \ln\left(\frac{64N(\epsilon, \delta/2)}{\sqrt{\delta}} \frac{\left(\sum_{j=1}^n 1/c_j\right)^2}{\left(\sum_{j=1}^n \beta_j/c_j\right)^2}\right) \\
+\frac{4\sum_{j=1}^n \beta_j \sum_{j=1}^n 1/c_j}{\sum_{j=1}^n \beta_j/c_j}
\end{aligned} \tag{33}
$$

From Inequality (30), we obtain an upper bound of the cost complexity of cProAL:

$$
(\sum_{j=1}^n |z_j| c_j) N(\epsilon, \delta/2) \quad \leq \quad \frac{(|\mathcal{Z}| + n) n N(\epsilon, \delta/2)}{\sum_{j=1}^n 1/c_j}
$$

which is further upper bounded by the following (34), combined with the upper bound on $|\mathcal{Z}|$ in (33)

$$
\begin{aligned}
&\left(32\frac{\sum_{j=1}^n 1/c_j}{\left(\sum_{j=1}^n \beta_j/c_j\right)^2} \ln\left(\frac{64N(\epsilon, \delta/2)}{\delta} \frac{\sum_{j=1}^n 1/c_j}{\sum_{j=1}^n \beta_j/c_j}\right)\right. \\
&\quad + \left.\quad \frac{4\sum_{j=1}^n \beta_j}{\sum_{j=1}^n \beta_j/c_j} + \frac{n}{\sum_{i=1}^n 1/c_j}\right) n N(\epsilon, \delta/2)
\end{aligned} \tag{34}
$$

by simple algebra. □

5 Aggregation of tProAL and cProAL

Neither tProAL nor cProAL is optimal. One dominates the other in different settings. Assume oracles have the same cost (spreading around the queries uniformly to all oracles). When only one oracle j^* has zero noise rate but all others have noise rate $1/2-\gamma$ ($0<\gamma \ll \frac{1}{\sqrt{n}}$), under same cost, the cost complexity of tProAL is

$$O(nc_{j^*} \ln\left(nN(\epsilon,\delta/2)/\delta\right) N(\epsilon,\delta/2))$$

while that of cProAL is

$$O\left(\frac{1}{\gamma^2}\ln\left(\frac{N(\epsilon,\delta/2)}{\delta\gamma}\right) N(\epsilon,\delta/2)\right)$$

for large n. In this case tProAL is better. With a skewed noise distribution, ST1 allows the good oracles to halt in a very few queries and stops the bad oracle wasting queries; while the single SeqHT integrating all oracles' quries with uniform weights run byST2 buries the advantage of the exceptionally good ones. However, when all oracles have same cost and same noise rate, cProAL saves a factor of n compared with tProAL. With a flat noise distribution, all oracle potentially take similar amount of queries to halt. The number queries it need in order to halt is n times that by the central-pool based SeqHT in ST2. Having no *a priori* knowledge of the the noise distribution, an aggregation of ST1 and ST2 may have advantage of both to some extent. The following proposed aggregation strategy can achieve the minimum of the cost complexities of the two (discussions on learning the noise distribution can be found in 6).

SequentialTest3(x, δ') (denoted as ST3)
Initialize the set of labels $z_j=\emptyset$ for $j=1,\cdots,n$
1. **do**
2. Select Oracle j that minimizes $(|z_j|+1)c_j$
3. Query Oracle j and get label l
4. $z_j \leftarrow z_j \cup \{l\}$
5. $p_j \leftarrow$ average of elements in z_j
6. $p \leftarrow$ average of elements in $\mathcal{Z}=\cup_{j=1}^n z_j$
7.

$$I_j \leftarrow \left[p_j - \sqrt{\frac{2\ln\left(4n|z_j|^2/\delta'\right)}{|z_j|}}, p_j + \sqrt{\frac{2\ln\left(4n|z_j|^2/\delta'\right)}{|z_j|}}\right]$$

8.

$$I \leftarrow \left[p - \sqrt{2\ln\left(\frac{4|\mathcal{Z}|^2}{\delta'}\right)/|\mathcal{Z}|}, p + \sqrt{2\ln\left(\frac{4|\mathcal{Z}|^2}{\delta'}\right)/|\mathcal{Z}|}\right]$$

9. **while** $0\in I$ and $0 \in I_j$
8. **if** $I \subset (-\infty,0)$ or $I_j \subset (-\infty,0)$, output -1
9. **else** output $+1$

ST3 is an aggregation of ST1and ST2. It has the same oracle-sampling strategy with ST1 and ST2. Each oracle runs a separate SeqHT based on the sample it gets. MeanwhileST3 maintains a central pool incorporating samples from all oracles and run a central SeqHT. Whoever either among the n-threaded SeqHTs or the central SeqH halts first, will return the correct label for $\mathbf{x}$. We thus split the failure probability δ' into two halves : one half corresponds to the event that none of the n individual SeqHTs halts, the other half corresponds to the event that the central-pool based single SeqHT does not halt. Using ST3 as SeqHTRoutine in ProAL, we obtain our thirds algorithm (denoted as tcProAL).

Theorem 3. $\exists k_1, k_2, k_3 \in (0, \infty)$ *such that the cost complexity of tcProAL is upper bounded by*

$$\min\left\{\min_{j=1,\cdots,n} \frac{k_1 c_j}{\beta_j^2} \ln\left(\frac{nN(\epsilon,\delta/2)}{\beta_j \delta/2}\right),\right.$$
$$\left(\frac{k_2 \sum_{j=1}^n 1/c_j}{\left(\sum_{j=1}^n \beta_j/c_j\right)^2} \ln\left(\frac{N(\epsilon,\delta/2)}{\delta/2} \frac{\sum_{j=1}^n 1/c_j}{\sum_{j=1}^n \beta_j/c_j}\right)\right.$$
$$\left.\left. + \frac{k_3 \sum_{j=1}^n \beta_j}{\sum_{j=1}^n \beta_j/c_j} + \frac{n}{\sum_{i=1}^n 1/c_j}\right)\right\} nN(\epsilon,\delta/2)$$

where $\beta_j = 1 - 2\alpha_j$. $N(\epsilon, \delta/2)$ *is the sample complexity for the realizable active learning algorithm.*

Proof Sketch. There are two parts coming together to form the total failure probability δ. tcProAL keeps same the part feeding into the realizable active learning algorithm **A** ($N(\epsilon, \delta/2)$ in the bounds kept same), and split in half the δ' handed to SeqHTRoutine : one half goes to ST1 and the other half goes to ST2. Theorem 1 and Theorem 2 proved correct for any δ', thus the same proofs holds if just replacing δ' by $\delta'/2$. By union bound, results of cost complexity for ST1 and ST2 will hold simultaneously. Whichever test halts first will be the correct answer. Therefore, the minimum of the two cost complexities gives the cost complexity for tcProAL. □

6 Future Directions

The reader may notice that the oracle sampling strategy in ST1 and ST2 relies solely on costs, independent of unknown noisy rates αs. It will be desirable to let knowledge of αs influence the choice of oracle. However, the reliability of oracles is typically unknown *a priori*. If the noise rate were known, it would be trivial to improve the selection criterion. Consider there are n oracles available for querying, with known cost c_j and known noise rate $\alpha_j \geq 0$, and $j = 1, \cdots, n$. We simply choose the oracle has $\min_j \frac{c_j}{(1/2-\alpha_j)}$; or formulate it as an optimization problem with noise rates and cost constraints.

With no prior knowledge about reliability, oracle reliability estimation can be very costly. In a single SeqHT, the expense for estimation of αs is no less than that of deciding the label. The hope is knowledge about the oracles can be acquired through the accumulation over a number of trials: feedback statistics can yield increasingly accurate estimations of αs without spending extra queries. In addtion we can record oracle statistics from the past SeqHTs and pass them to the future calls as priors on oracle accuracy estimation to help oracle selection.

We can potentially modify ProAL and ST1 in the following way: First, let ST1 also output the sample mean p^i_j for $j = 1, \cdots, n$, and the halting oracle index. Here i stands for the ith example in querying. Second, ProAL will update the sample mean by $p^{i+1}_j = \sum_i p^i_j$, and pass these statistics to the future call of ST1. Third, ST1 incorporates $\hat{\alpha}_j = (1 - p^{i+1}_j)/2$, for $j = 1, \cdots, n$ into oracle selection criterion (Step 2 in ST1): always choose

$$j^* = \arg\min_{j=1,\cdots,n} \frac{(|z_j| + 1)c_j}{(1/2 - \hat{\alpha}_j)^2}$$

The more examples ProAL experiences, the better it estimates the noise rate of each oracle, and which oracles halt first more frequently. It can thus generate accumulated wisdom in choosing oracles, assuming the same or at least some common oracles are availble across learning epochs.

Due to the reduction technique, SeqHTRoutine has no control on the upcoming example: which is decided by a black-box realizable active learning algorithm **A**. The modified ProAL can be formulated as an online learning procedure about oracle reliability, involving exploration vs. exploitation tradeoff. Also left for future exploration is a regret bound analysis for online learning about oracles: the regret arises from not always choosing the presumably best oracle.

References

Angluin, D., & Larid, P. (1987). Learning from noisy examples. *Machine Learning*, *2*, 343–370.

Balcan, M.-F., Beygelzimer, A., & Langford, J. (2006). Agnostic active learning. *In Proc. of the 23rd International Conference on Machine Learning (ICML)*.

Bar-Yossef, Z. (2003). Sampling lower bounds via information theory. *In Proc. of the 35th Annual ACM Symposium on the Theory of Computing*.

Dasgupta, S. (2005). Coarse sample complexity bounds for active learning. *In Proc. of Neural Information Processing Systems (NIPS)*.

Donmez, P., & Carbonell, J. (2008). Proactive learning: Cost-sensitive active learning with multiple imperfect oracles. *In Proc. of the 17th ACM Conference on Information and Knowledge Management*.

Donmez, P., Carbonell, J. G., & Schneider, J. (2009). Efficiently learning the accuracy of labeling sources for selective sampling. *In Proc. of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*.

Hanneke, S. (2007). Teaching dimension and the complexity of active learning. *In Proc. of the 20th Annual Conference on Learning Theory (COLT)*.

Hanneke, S. (2009). *Theoretical foundations of active learning*. Doctoral dissertation, Carnegie Mellon University.

Kääriäinen, M. (2006). Active learning in the non-realizable case. *In Proc. of the 17th International Conference on Algorithmic Learning Theory*.

Laird, P. D. (1988). *Learning from good and bad data*. Norwell, MA, USA: Kluwer Academic Publishers.

Vidyasagar, M. (2003). *Learning and generalization: With applications to neural networks (second edition)*. Networks: Springer-Verlag.

Wald, A. (1945). Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, *16*, 117–186.

MACHINE LEARNING
DEPARTMENT
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213