# Adaptive Proactive Learning with Cost-Reliability Tradeoff

Liu Yang      Jaime Carbonell

**Carnegie Mellon**

ML

**MACHINE LEARNING**
**D E P A R T M E N T**

# Adaptive Proactive Learning with
# Cost-Reliability Tradeoff

## Liu Yang      Jaime Carbonell

December 2009
CMU-ML-09-114

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

## Abstract

Proactive Learning is a generalized form of active learning where the learner must reach out to multiple oracles exhibiting different costs and reliabilities (label noise). One of the its major goals is to capture the cost-noise tradeoff in oracle selection. Sequential active learning exhibits coarse accuracy at the beginning and progressively refine prediction at later stages. The ability to learn oracle accuracies over time and select better oracles or oracle ensembles lead to potentially faster error reduction rate as a function of total cost, and thus improve its cost complexity. To realize this potential, we propose a statistical model that adapts to a range of accuracies at different stages of active learning. In a more general scenario, we formulate the problem as maximum submodular coverage subject to a budget envelope.

# 1 Proactive Learning with Persistent Oracles

Consider the binary classification task. Let $\mathcal{F}$ be the hypothesis class. Let $\mathcal{X}$ represent the data space, and $\mathcal{Y} = \pm 1$ represent the label space. The target function $f^* \in \mathcal{F}$. Let $I = \{1, 2, \cdots, n\}$. We consider the active learning problem with $n$ persistent oracles whose noise is arbitrary. The cost for oracle $j$ is $c_j > 0$, $j \in I$. We have the following assumptions on the oracle error rates.

**Assumption 1.** *The error indicator $\mathbb{1}(O_i(X) \neq f^*(X))$ of individual oracles are independent, where $O_i(X)$ is the answer of Oracle $i$ for a given example $X$ and $i = 1, \cdots, n$.*

Given $n$ oracles with varied noise rates and costs, the goal of Proactive learner is to choose an ensemble of oracles with minimum cost and whose error rate is no greater than $\frac{\epsilon}{2}$ as the labeling mechanism, and 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 the oracles we study in this work are persistent : they provide the same answer for a given example if asked multiple times. Furthermore, the situation with arbitrary noise forces one to sample the oracle space, since there is no theoretical guarantee that one can get better than $\frac{\epsilon}{2}$ error rate by querying the same oracle multiple times with different examples.

We propose a meta-procedure that takes any agnostic active learning algorithm **A** as subroutine. The agnostic algorithm **A** halts and outputs a classifier after making certain label requests. The meta-procedure lets **A** choose examples to query, and hands back to **A** the "true" label after calling the oracle selection routine and synchronise the answers from the selected oracles by (weighted) majority vote. Furthermore, we may choose an ensemble of oracles adaptively to accommodate a range of accuracy: different ensembles for different level of accuracy. Or we may pick the ensemble offline and to use it for all queries. Our later analysis shows that the former can produce a modest cost savings over the latter simpler method.

# 2 Combining Oracle Answers by Weighted Majority Vote

We combine the answers from the set of selected oracles $S$, by (Weighted) Majority Vote. Consider first a simple majority vote on the label of example $X$. Let $Y$ be the true label, and $Y_i$ be the answer by oracle $i$. Denote the majority vote error rate as $er_{\mathrm{maj}}$, and the average error rate of $n$ oracles as $\bar{\epsilon}_S = \sum_{i \in S} \frac{\epsilon_i}{|S|}$. Assumption 1 implies that $er_{\mathrm{maj}}(S)$ should exponentially decrease as a function of the number of oracles. By Hoeffding inequality and Assumption 1, after querying a sample size of $m$,

$$
\begin{aligned}
er_{\mathrm{maj}} &= \mathbb{P}\left(\frac{\sum_{i \in S} \mathbb{1}(Y_i \neq Y)}{|S|} \geq \frac{1}{2}\right) \\
&= \mathbb{P}\left(\frac{\sum_{i \in S} \mathbb{1}(Y_i \neq Y)}{|S|} - \bar{\epsilon} \geq \frac{1}{2} - \bar{\epsilon}\right) \\
&\leq \exp\left(-2|S|(\frac{1}{2} - \bar{\epsilon})^2\right)
\end{aligned}
\tag{1}
$$

Thus $er_{\text{maj}} < \frac{\epsilon}{2}$, sufficient if

$$\bar{\epsilon} < \frac{1}{2} - \sqrt{\frac{1}{2m} \ln \frac{2}{\epsilon}} \tag{2}$$

Let us extend the above analysis to Weighted Majority Vote. Denote the weighted majority vote error rate as $er_{\text{wmaj}}$. Denote $w_i$ for $i \in S$ as the weights of a subset of oracles we choose. $\bar{\epsilon}_S = \sum_{i \in S} \frac{\epsilon_i w_i}{|S|}$. We determine the weights by minimizing $er_{\text{wmaj}}$. By Hoeffding inequality,

$$
\begin{aligned}
er_{\text{wmaj}} &= \mathbb{P}\left( \frac{\sum_{i \in S} w_i \mathbb{1}(Y_i \neq Y)}{|S|} \geq \frac{\sum_{i=1}^{n} w_i}{2|S|} \right) \\
&= \mathbb{P}\left( \frac{\sum_{i \in S} w_i \mathbb{1}(Y_i \neq Y)}{|S|} - \bar{\epsilon}_S \geq \frac{\sum_{i=1}^{n} w_i}{2|S|} - \bar{\epsilon}_S \right) \\
&\leq \exp\left( -\frac{2|S|^2 (\frac{\sum_{i=1}^{n} w_i}{2|S|} - \bar{\epsilon}_S)^2}{\sum_{i \in S} w_i^2} \right) \\
&\leq \exp\left( -\frac{2|S|^2 \left( \frac{\sum_{i \in S} w_i}{2|S|} - \frac{\sum_{i \in S} \epsilon_i w_i}{|S|} \right)^2}{\sum_{i \in S} w_i^2} \right) \\
&= \exp\left( -\frac{2(\sum_{i \in S} w_i(\frac{1}{2} - \epsilon_i))^2}{\sum_{i \in S} w_i^2} \right)
\end{aligned}
\tag{3}
$$

To minimize this bound on the error rate, we set the partial derivative of equation (3) to be zero

$$\frac{\partial \left( \frac{(\sum_{i \in S} w_i(1/2 - \epsilon_i))^2}{\sum_{i \in S} w_i^2} \right)}{\partial w_i} = 0$$

and get

$$w_i = (1/2 - \epsilon_i) \frac{\sum_{i \in S} w_i^2}{\sum_{i \in S}(1/2 - \epsilon_i)w_i} \tag{4}$$

If error rate $\epsilon_i$ or its estimation is known, we can calculate $w_i$s by Equation (4). If $\epsilon_i$ is unknown, we simply use majority vote. We also notice that when $\epsilon_i = 1/2$, $w_i = 0$. Equation (4) assigns zero weights to those oracles whose noise rate amounts to random guess. However, for $\epsilon_i = 0$ (a perfect oracle), Equation (4) does not gives an especially large $w_i$, due to the relaxation of the original $er_{\text{wmaj}}$ by Hoeffding inequality. To this end, we denote **OrSelRoutine** $(\epsilon)$ as any oracle selection procedure that chooses a min-cost subset of oracles $S \subseteq I$, such that with weights $w_i$ for $i \in S$ calculated as in Equation (4), $er_{\text{wmaj}} \leq \frac{\epsilon}{2}$. In the later sections, we will provide efficient optimization procedures to this task.

# 3  Oracle Selection Adaptive to Active Learning

If there is any hope that switching oracles during active learning can improve the cost complexity, then the following model should realize that potential.

2

**Assumption 2.** *The availability of oracles.* $\exists$ *function* $g : \epsilon \mapsto c$, *mapping from a required accuracy to the cost per example, such that* $\forall \epsilon$, $\exists$ *oracle whose cost is* $g(\epsilon)$, *and error rate is* $\epsilon$.

According to Assumption 2, for a given accuracy, there is at least an oracle with a certain available noise rate. The earlier stage of active learning needs a relatively coarse accuracy, however, higher accuracies should be reached in later stages. This can be modeled by a sequential level of accuracy

$$\epsilon_1 = 1/2, \epsilon_2 = 1/4, \epsilon_3 = 1/8, \cdots, \epsilon_t = (1/2)^t, \cdots$$

Intuitively, having options to select the oracles may let active learning have a faster error reduction rate in terms of the cost. The idea of adaptive labeling mechanism construction is: given the accuracy level $\epsilon_t$ of active learning, we choose an ensemble of $M_t$ oracles to form labeling mechanism to accommodate $\epsilon_t$. The lower the $\epsilon_t$, the larger $M_t$. Suppose $m_t$ is the sample complexity of **A** given $\epsilon_t$. We have the following algorithm for proactive learning with adaptive oracle selection.

**Proactive Learner with Adaptive Oracle Selection** (denoted as **AdaProAL**)
Input an agnostic active learning algorithm **A**
0. Initialize $t = 1$
1. **do**
2.     $\epsilon_t \leftarrow (1/2)^t$
3.     $(S_t$ and $w_j^t$ for $j \in S_t) = $ OrSelRoutine $(\epsilon_t)$
4.     $j = 0$
5.     **do**
6.         Let **A** choose a query point $X$ from the unlabeled data
7.         $Y = \text{sgn}(\sum\limits_{j \in S_t} w_j^t f_j(X))$
8.         $j = j + 1$
9.         Return $Y$ to **A**
10.     **until** $j = m_t$
11.     $t = t + 1$
12. **until A** halts

Alternatively, the non-adaptive approach constructs the labeling mechanism ahead of time and uses that chosen mechanism all the time while running the active learning algorithm.

**Agnostic Proactive Learning with Non-Adaptive Oracle Selection**
Input an agnostic active learning algorithm **A**
0. $(S$ and $w_j$ for $j \in S) = $ OrSelRoutine $(\frac{\epsilon}{2})$
1. Initialize $t = 0$
2. **do**
3.     $t = t + 1$
4.     Let **A** choose a query point $X_t$ from unlabeled data
5.     Let $y = \text{sgn}(\sum\limits_{j \in S} w_j f_j(X_t))$
6.     Return $y$ into **A**

7. **until A** halts

Denote $\theta$ as the disagreement coefficient [Hanneke, 2007] and $d$ as VC-dimension [Vapnik, 1998]. The upper bound of sample complexity in achieving a given level of accuracy $O(\theta\epsilon_t)$ is

$$\tilde{O}\left(\theta^2\left(d + \log\left(\frac{1}{\delta}\right)\right)\right)$$

according to a step of the proof in [Hanneke, 2007], if we choose the $\mathbf{A^2}$ algorithm given in [Balcan et al., 2006] as the active learning algorithm **A**.

Thus the cost complexity of the non-adaptive approach is

$$\tilde{O}\left(\theta^2\left(d + \log\left(\frac{1}{\delta}\right)\right)\right) g\left(\frac{\epsilon}{2}\right)\log\left(\frac{\theta}{\epsilon}\right)$$

whereas the cost complexity of AdaProAL is

$$\tilde{O}\left(\theta^2\left(d + \log\left(\frac{1}{\delta}\right)\right)\right)\sum_{i=1}^{\lceil\log\left(\frac{32\theta}{\epsilon}\right)\rceil} g(2^{-i})$$

Note $\mathbf{A^2}$ achieves the accuracy $16\theta 2^{-i}$ by making $\theta d^2$ queries. We want the noise rate of the last oracle to be $< \epsilon$; thus we let

$$16\theta 2^{-i} = \epsilon/2$$

Therefore $i = \log(\frac{32\theta}{\epsilon})$.

# 4 Proactive Learning with Cost-Reliability Assumption

The cost-reliability tradeoff assumption that more reliable oracles cost more than noisy ones, might be formalized as

**Assumption 3.** $\exists\, \beta > 0, \gamma > 0$ *s.t. for* $i \in I$,

$$c_i\epsilon_i^\gamma \le \beta \tag{5}$$

Large cost $c_i$ leads to a small error rate $\epsilon_i$. Large error rate $\epsilon_i$ drives the cost down. Our algorithm is $(\alpha, \beta)$-dependent. It will be interesting to explore algorithms that adapts to the value of $\alpha$ and $\beta$. When $\gamma < 1$, a decrease of $\epsilon_i$ has to be how much faster than the increase of the cost $c_i$, as $\epsilon_i^\gamma$ is sublinear; whereas with $\gamma > 1$, the increase of $\epsilon_i$ forces a faster reduction on cost $c_i$.

A trivial labeling mechanism is to pick a single oracle whose cost $\ge \beta\left(\frac{2}{\epsilon}\right)^\gamma$. Condition (5) will force its error rate to be $< \frac{\epsilon}{2}$. However, the hope is that an ensemble of cheap oracles can have just as good accuracy as the expensive one at lower cost. Based on Assumption 3, we will provide an algorithm that requires zero query to construct a label mechanism whose error rate $< \frac{\epsilon}{2}$, if the upper bound in (5) is tight.

## 4.1 The Algorithm and Complexity Analysis

Given the error rate $\frac{\epsilon}{2}$ set up by the adversary, we can choose a cost $c$, and an ensemble of $M$ oracles with roughly this cost (or within a factor of 2 difference) by Assumption 2. The goal is to minimize the total cost of the chosen ensemble, subject to Inequality 2 and Assumption 3. We formulate this task as the following optimization problem:

$$\min \quad cM$$
$$\text{s.t.} \quad \left(\frac{\beta}{c}\right)^{1/\gamma} = \frac{1}{2} - \sqrt{\frac{1}{2M} \ln \frac{2}{\epsilon}}$$

Since

$$M = \frac{\ln(2/\epsilon)}{2\left(\frac{1}{2} - (\frac{\beta}{c})^{1/\gamma}\right)^2}$$

we set

$$\frac{\partial\,(cM)}{\partial c} = \frac{\partial\left(\frac{c\ln(2/\epsilon)}{2\left(\frac{1}{2}-(\frac{\beta}{c})^{1/\gamma}\right)^2}\right)}{\partial c} = 0$$

Thus

$$c^* = \beta 2^\gamma (1 + \frac{2}{\gamma})^\gamma \tag{6}$$

$$M^* = \left\lceil 2\ln\left(\frac{2}{\epsilon}\right)\left(1 + \frac{\gamma}{2}\right)^2 \right\rceil \tag{7}$$

We notice that $c^*$ has nothing to do with $\epsilon$, and is fixed once $\gamma$ and $\beta$ are set up. $M^*$ is inversely related to $\epsilon$ : the higher level of accuracy the active learning algorithm requires, the more oracles will be needed to accommodate such an accuracy. We combine the answers from the $M^*$ selected oracle by a simple majority vote (weighted vote by estimating oracle accuracy may yield even better results).

**Oracle Selection Routine with Assumption 3** ($\frac{\epsilon}{2}$) (denoted as OrSelRoutineA3)
1. Calculate $M^*$ by (7)
2. Calculate $c^*$ by (6)
3. $S = \{j : j \in I \wedge c_j \approx c^*\}$ with size $M^*$
4. Output $S$

**Algorithm 1**
Choose the $\mathbf{A^2}$ algorithm given in [Balcan et al., 2006] as the algorithm $\mathbf{A}$
0. Initialize $t = 0$
1. **do**
2.    $t = t + 1$

3.    Let **A** choose a query point $X_t$ from unlabeled data

4.    $S_t = \text{OrSelRoutineA3} \left( \frac{\epsilon_t}{2} \right)$

5.    Let $y = \text{sgn}( \sum_{j \in S_t} f_j(X_t))$

6.    Return $y$ into **A**

7. **until A** halts

**Theorem 1.** *The total cost complexity of Algorithm 1 is*

$$\tilde{O} \left( \theta^2 \left( d + \log \left( \frac{1}{\delta} \right) \right) \right) c^* M^*$$

## 4.2   Adaptive Oracle Selection Saves A Constant Factor

To illustrate the range of saving one can get by using the adaptive oracle selection procedure com-pared to the non-adaptive version, we suppose $\tilde{O} \left( \theta^2 \left( d + \log_2 \left( \frac{1}{\delta} \right) \right) \right)$ as previously mentioned is a constant $b$, independent of $t$. For instance, under threshold classifier, the disagreement coefficient $\theta = 2$ and VC-dim $d = 1$; thus $\tilde{O} \left( \theta^2 \left( d + \log_2 \left( \frac{1}{\delta} \right) \right) \right)$ is a constant. Define $M_t^*$ as the number of oracles selected to accommodate $\epsilon_t$. For adaptive oracle selection, Algorithm 1 has $g(\epsilon) = cM(\epsilon)$. By equation (7), the cost complexity of using the adaptive procedure is

$$bc^* \sum_{t=1}^{\lceil \log_2 \left( \frac{16\theta}{\epsilon} \right) \rceil} M_t^* \left( \frac{\epsilon_t}{2} \right) \quad = \quad bc^* \sum_{t=1}^{\lceil \log_2 \left( \frac{16\theta}{\epsilon} \right) \rceil} \lceil 2 \ln \left( \frac{1}{\epsilon_t} \right) \left( 1 + \frac{\gamma}{2} \right)^2 \rceil \tag{8}$$

where

$$\sum_{t=1}^{\lceil \log_2 \left( \frac{16\theta}{\epsilon} \right) \rceil} \ln \left( \frac{1}{\epsilon_t} \right) = \left( \frac{\log \left( \frac{16\theta e}{\epsilon} \right)}{2} \right) \approx \frac{\log_2^2 \left( \frac{16\theta e}{\epsilon} \right)}{4e^2} \tag{9}$$

The total sample complexity of the active learning is $b \log_2 \left( \frac{1}{\epsilon} \right)$.

The cost complexity for the non-adaptive approach is

$$bc^* M^* \left( \frac{\epsilon}{2} \right) \log_2 \left( \frac{1}{\epsilon} \right) \quad = \quad bc^* \lceil 2 \ln \left( \frac{2}{\epsilon} \right) \left( 1 + \frac{\gamma}{2} \right)^2 \rceil \log_2 \left( \frac{1}{\epsilon} \right) \tag{10}$$

Comparing (9) and (10), for small $\theta$, we have

$$\frac{\log_2^2 \left( \frac{16\theta e}{\epsilon} \right)}{4e^2} < k \log_2^2 (\frac{1}{\epsilon})$$

with $k > 1$. Thus the adaptive oracle selection procedure saves the cost complexity by a constant factor compared with the non-adaptive version, under the above specified scenario.

Empirical evidence from [Donmez & Carbonell, 2008] is consistent with the above example: picking oracles adaptively can reduce the cost complexity by a constant factor versus always using the better and expensive oracle or always using the cheaper less reliable one. Figure 1 and Figure 2 display the trends of classification error as a function of the total cost on the Adult dataset and the VY-Letter dataset, respectively. It studies a proactive learning scenario where there are two oracles: one is cheap but noisy, the other expensive but reliable. Each plot in Figure 1 indicates a different cost ratio between the two oracles (same for Figure 2). Experiments on the two datasets show that, when the classification accuracy is low, the proactive learner tends picks the low-cost oracle; howevter, it tends to select the high-cost oracle once the error rate has been significant reduced. At the later stages, the curve of the proactive learner goes roughly in parallel with that of the baseline, meaning their speed of error rate reduction is roughly the same. If we draw a horizontal line on the plot, the amount of total cost in achieving certain error rate by the proactive learner is roughly a half of that by the baseline. Thus one roughly saves a factor of 2 by using the adaptive oracle selection. For these two datasets, the constant factor can be smaller or larger depending on the difficulty of the classification task. This paper provides the theoretical framework for proactive learning, and our analysis is quite consistent with the above-mentioned empirical results.
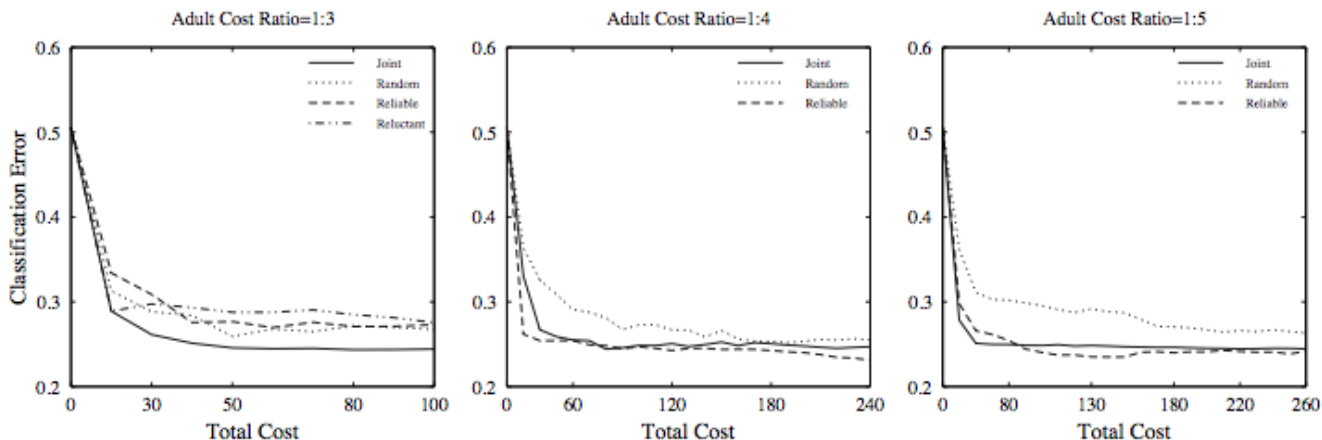


Figure 1: Performance Comparison on the Adult dataset. The cost ratio is indicated above each plot.

# 5 Maximum Submodular Coverage Subject to a Budget : A More General Scenario

If we do not make any explicit assumption on cost-reliability tradeoff, that is, cost and accuracy are not in strictly monotonic inverse relation, the proactive learning problem is actually finding a subset of oracles with small enough majority vote error rate, given a budget $B$. It can be formulated
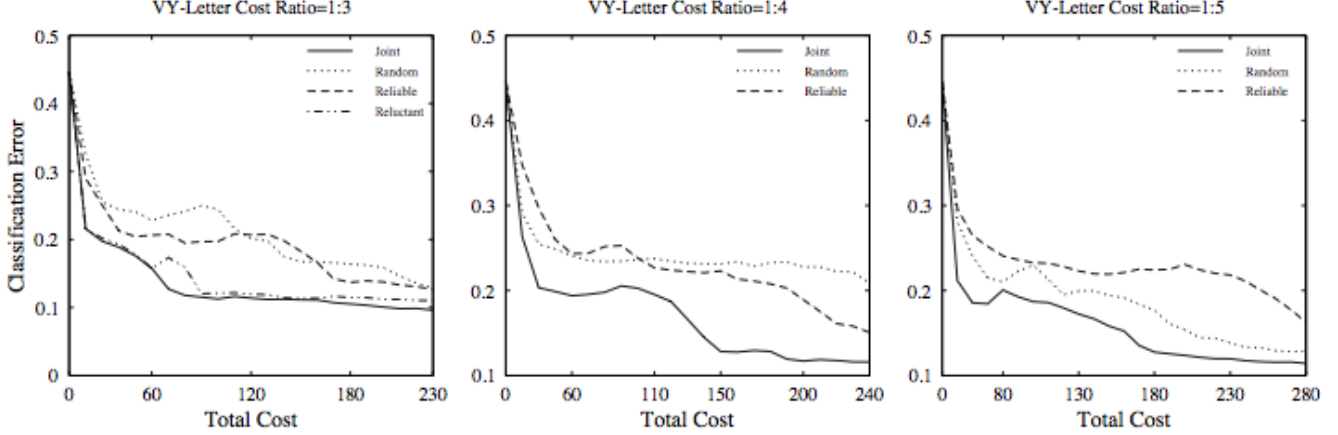
Figure 2: Performance Comparison on the VY-Letter dataset. The cost ratio is indicated above each plot.

as the following optimization problem :

$$\max_{S \in I}\{f(S) : \sum_{i \in S} c_i \leq B\} \qquad (11)$$

where the majority-vote accuracy $f : S \mapsto \mathcal{R}$ with $S \subseteq I$ is defined as :

$$
\begin{aligned}
f(S) &= E\mathbb{1}\left(\left(\frac{\sum_{i \in S} w_i(Y_i \neq Y)}{\sum_{i \in S} w_i}\right) < 1/2\right) \qquad (12) \\
&= \frac{1}{|\mathcal{Z}|} \sum_{(X,Y \in \mathcal{Z})} \mathbb{1}\left(\left(\frac{\sum_{i \in S} w_i \mathbb{1}(Y_i \neq Y)}{\sum_{i \in S} w_i}\right) < 1/2\right) \qquad (13)
\end{aligned}
$$

where $Y$ is the true label and $Y_i$ is the answer by $i$th oracle. $f$ has the following properties. First, $f$ is non-decreasing, and polynomial computable set function. Second, $f$ is monotonic, since $\forall S \subseteq T$, we have $f(S) \leq f(T)$. Third, $f$ is submodular: it increases more by adding elements to a small set, than by adding to a super set. $\forall S, T \in I$, we have $f(S) + f(T) \leq f(S \cup T) + f(S \cap T)$. The role of submodularity plays for set functions, is similar to that of concavity for ordinary functions.

Problem (11) is the problem of Maximum Submodular Coverage Subject to a Budget. [Sviridenko, 2004] describes a greedy algorithm for this type of problem, as an $(1 - e^{-1})$ approximation algorithm for maximizing a nondecreasing submodular set function subject to a knapsack constraint. The quality of greedy solutions is strongly related to submodularity of the set function. When the submodularity property holds (as in our case), the number of computations necessary to get a greedy solution can be significantly reduced. The following greedy approximation algorithm **bMaxSubCover** efficiently solves Problem (11).

8

**bMaxSubCover**(B)

0. $I = \{1, 2, \cdots, n\}$

1. Phase 1: $S_1 = \arg\max_{|S|=1,2} f(S)$

2. Phase 2: for every $U \subseteq I$ s.t. $|U| = 3$

3.     Initialize $S^0 = U$, $k = 0$

4.     **do**

5.        $\forall e_j \in I/S^k$, compute

$$\Delta^k(e_j) = \frac{f(S^k + e_j) - f(S^k)}{c_i}$$

6.

$$e_{j0} = \underset{\left(e_j \in I/S^k\right) \wedge \left(c_i \leq B - \sum_{e_j \in S^k} c_j\right)}{\arg\max} \Delta^k(e_j)$$

7.       $S_K \leftarrow S_k \cup \{e_{j0}\}$

8.     **while**($e_{j0}$ exists)

9. $S_2 \leftarrow S^k$ as local optimal obtained by Phase 2

10. If $f(S_1) \geq f(S_2)$ output $S_1$, otherwise output $S_2$

    The algorithm has $\alpha$ performance guarantee ($0 < \alpha < 1$), if it always outputs a solution of value that is not smaller than $\alpha$ times the value of the optimal solution. The following performance guarantee of **bMaxSubCover** is due to [Sviridenko, 2004].

**Theorem 2.** *The worst-case performance guarantee of the above greedy algorithm bMaxSubCover for solving Problem (11) is $(1 - e^{-1}) \approx 0.632$. In another word, assume $S$ is the subset output by bMaxSubCover(B), the following holds:*

$$f(S) \leq (1 - e^{-1})f(S^*)$$

*where $S^*$ is the solution found by the exact approach.*

    Without knowing the smallest amount to spend in letting $f(S) > 1 - \frac{\epsilon}{2}$, a double-and-guess on the budget $B$ can help decide the minimum budget.

**Subroutine1** ($\epsilon_i$ for $i = 1, \cdots, n$)

0. Initialize $B = 1$

1. **do**

2.   $S \leftarrow$bMaxSubCover(B, $\epsilon_i$ for $i = 1, \cdots, n$)

3.   $B \leftarrow 2B$

4. **while** ($f(S) < 1 - \frac{\epsilon}{2}$)

5. Output $S$ and $B$.

# 6    Conclusion

This paper provides the theoretical framework for proactive learning. We propose a meta-procedure for the active learning problem with multiple persistent oracles under arbitrary noise. Having options to select oracles may let active learning have a faster error reduction rate as a function of the total cost; we thus choose subsets of oracles adaptive to a range of accuracies. The non-adaptive approach is to construct the labeling mechanism ahead of time and use that all the time while running the active learning algorithm. Analysis in some specified scenarios shows the adaptive oracle selection procedure saves the cost complexity by a constant factor compared with the non-adaptive version, and our analysis is quite consistent with empirical results from [Donmez & Carbonell, 2008]. We further combine the answers from the set of selected oracles by Weighted Majority Vote.

Under the assumption that more reliable oracles cost more than noisy ones, we provide an algorithm that cost zero query to construct a minimum-cost label mechanism whose error rate $< \frac{\epsilon}{2}$. Without assuming that cost and accuracy are not in strictly monotonic inverse relation, we formulate the problem as maximum submodular coverage subject to a budget, that can be solved by a greedy algorithm with $1 - e^{-1}$ worst-case performance guarantee.

# References

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

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*.

Hanneke, S. (2007). A bound on the label complexity of agnostic active learning. *Proceedings of the 24th international conference on Machine learning*.

Sviridenko, M. (2004). A note on maximizing a submodular set function subject to knapsack constraint. *Operations Research Letters*, *32*, 4143.

Vapnik, V. N. (1998). *Statistical learning theory*. Wiley-Interscience.

**ML**

**MACHINE LEARNING**
**D E P A R T M E N T**

Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213

**Carnegie Mellon®**