

Active Learning by Learning

Wei-Ning Hsu

Department of Electrical Engineering,
National Taiwan University
mhng1580@gmail.com

Hsuan-Tien Lin

Department of Computer Science and Information Engineering,
National Taiwan University
htlin@csie.ntu.edu.tw

Abstract

Pool-based active learning is an important technique that helps reduce labeling efforts within a pool of unlabeled instances. Currently, most pool-based active learning strategies are constructed based on some human-designed philosophy; that is, they reflect what human beings assume to be “good labeling questions.” However, while such human-designed philosophies can be useful on specific data sets, it is often difficult to establish the theoretical connection of those philosophies to the true learning performance of interest. In addition, given that a single human-designed philosophy is unlikely to work on all scenarios, choosing and blending those strategies under different scenarios is an important but challenging practical task. This paper tackles this task by letting the machines adaptively “learn” from the performance of a set of given strategies on a particular data set. More specifically, we design a learning algorithm that connects active learning with the well-known multi-armed bandit problem. Further, we postulate that, given an appropriate choice for the multi-armed bandit learner, it is possible to estimate the performance of different strategies on the fly. Extensive empirical studies of the resulting ALBL algorithm confirm that it performs better than state-of-the-art strategies and a leading blending algorithm for active learning, all of which are based on human-designed philosophy.

1 Introduction

Active learning is a machine learning setup that enables machines to cleverly “ask questions” to reduce the amount of labeling efforts (Settles 2010). The vast majority of research work on active learning is focused on transforming the human philosophy of asking questions into programmable strategies (Tong and Koller 2002; Donmez and Carbonell 2008; Huang, Jin, and Zhou 2010). When the philosophy happens to match the characteristics of the data set on hand, the corresponding strategy can result in promising practical performance. However, there are many different philosophies behind different “human-asked” questions, and no single philosophy is likely to satisfy the characteristics of every data set. For any given data set, properly choosing the strategies is thus an important practical task.

Consider the scenario when we were children. We were not commanded to ask questions based on a single philosophy. Instead, it is our nature to experiment with several different (perhaps given) philosophies, and gradually determine the advantages and disadvantages of each philosophy in different situations by evaluating our learning performance along the way. That is, we *learn* our own powerful strategy as we grow, rather than merely implementing one fixed strategy. In other words, we conduct *active learning by learning*, as opposed to *active learning by acting*.

In this paper, we study how the machines can likewise conduct *active learning by learning*, instead of merely acting with a single human-designed strategy. We consider a practical task that mimics our childhood learning scenario: letting the machine learn to adaptively choose among some existing strategies based on their estimated contributions to the performance measure of interest. We connect the task to the multi-armed bandit problem for adaptive decision making (Beygelzimer et al. 2011). Consequently, we propose a novel approach for active learning that involves modifying a state-of-the-art method to solve the problem, and deriving a reward scheme that closely relates to the performance measure of interest. The proposed approach, shorthanded ALBL for active learning by learning, essentially results in a clever probabilistic blending of the strategies subject to their time-varying performance on the given data set, and echoes adaptive blending approaches for other machine learning problems (Jacobs et al. 1991).

We also empirically compare ALBL with several human-designed strategies, and demonstrate that ALBL is indeed able to use the derived reward to adaptively choose the best strategy and therefore achieve the best performance. Furthermore, experimental results show that ALBL is not only significantly better than a naive blending approach, but is also often better than the state-of-the-art adaptive blending approach COMB (Baram, El-Yaniv, and Luz 2004), which is based on a human-designed criterion during blending. The results indicate that ALBL is a superior choice for blending human-designed strategies adaptively.

The remainder of this paper is organized as follows. Section 2 outlines the active learning problem and reviews related work. Section 3 presents and illustrates the proposed ALBL approach. We discuss experimental results in Section 4 and conclude in Section 5.

2 Related Work

Active learning can be divided into two categories: stream-based and pool-based. In stream-based active learning, each instance is drawn from some distribution in a streaming manner and the learner has to decide immediately whether to query the label of this instance or not. Although their data access is more restricted, stream-based active learning algorithms (Cohn, Atlas, and Ladner 1994; Chu et al. 2011; Beygelzimer, Dasgupta, and Langford 2009) generally come with more solid theoretical performance guarantees.

Pool-based active learning (Settles 2010) is a more realistic setup that allows more flexible access to data and will be the focus of this paper. In pool-based active learning problems, a learner is presented with an unlabeled pool and a labeled pool in the beginning. We denote the whole data pool of n instances by $D = \{x_1, x_2, \dots, x_{n_u}, (x_{n_u+1}, y_{n_u+1}), \dots, (x_n, y_n)\}$, where the input instances $x_i \in R^d$ and y_i is the label of x_i . The pool D is the union of the unlabeled pool D_u that contains the first n_u instances, and the labeled pool D_l of the other $n - n_u$ instances along with their labels. The initial D is assumed to be generated i.i.d. from some distribution $P(x, y)$ before the labels y_i are hidden to form D_u . Note that in general, we can only access a small D_l initially, while the unlabeled pool D_u is relatively large.

Given the initial D_l , the learner trains a classifier f_0 . During the iterations $t = 1, \dots, T$, by considering D_l , D_u , and f_{t-1} , the learner selects one instance $x_j \in D_u$ to query its label. This instance is then moved to the labeled pool, and the learner can train a new classifier f_t with the updated D_l . With a small query budget T , the goal of active learning is to maximize the average test accuracy of those f_t , where the test accuracy can be defined from a separate test set that is also sampled from $P(x, y)$ like D .

Existing works on pool-based active learning predominantly focus on establishing reasonable criteria for selecting which instance to label. One popular criterion is called uncertainty sampling (Lewis and Gale 1994), which queries the instance that is most uncertain to the classifier f_{t-1} . For example, Tong and Koller (2002) propose to query the instance closest to the decision boundary of the SVM-trained f_{t-1} (Vapnik 1998). Uncertainty sampling assumes that f_{t-1} only needs fine-tuning around the boundary, and thus can be less satisfactory if f_{t-1} is not good enough.

Representative sampling resolves the caveats of uncertainty sampling by taking both uncertainty and representativeness of an instance into account. Researchers have proposed a wide variety of measurements for the representativeness. For example, Nguyen and Smeulders (2004) and Donmez, Carbonell, and Bennett (2007) claim that an instance around the boundary is more representative if it resides in a denser neighborhood, and propose a density-weighted criterion. Another route for measuring representativeness is through clustering. The PSDS approach (Donmez and Carbonell 2008) establishes a distance function through clustering to estimate the representativeness of instances. On the other hand, Dasgupta and Hsu (2008) propose a more sophisticated hierarchical clustering view, and use the cluster information to calculate representativeness. Some another

work like the QUIRE approach (Huang, Jin, and Zhou 2010) measures the representativeness by estimating the possible label-assignments for unlabeled instances.

However, there are usually no strong connections between the human-designed criterion and the performance measure of interest. In addition, what a human believes to be good questions may not work well on every data set and every situation. This deficiency hints the need for choosing from several different algorithms in a data-dependent and adaptive manner. One existing solution is called COMB (Baram, El-Yaniv, and Luz 2004), which will be further discussed later in Section 3.3 after we introduce our solution.

3 The Proposed Approach

Our solution is a novel approach called Active Learning by Learning (ALBL). The approach solves the task of choosing from a candidate set of existing algorithms adaptively based on their estimated contributions to the learning performance on a given data set. Our design is based on a well-known adaptive learning problem called the multi-armed bandit problem (Robbins 1985; Vermorel and Mohri 2005). Next, we introduce the multi-armed bandit problem and connect it to the task above.

The multi-armed bandit problem simulates what a gambler would do in a casino. Assume that the gambler is given K bandit machines and a budget of T iterations. The gambler is then asked to sequentially decide which machine to pull in each iteration $t = 1, \dots, T$. On being pulled, the bandit machine randomly provides a reward from a machine-specific distribution unknown to the gambler. The goal of the gambler is to maximize the total rewards earned through the sequence of decisions. To earn the most rewards, the gambler typically has to consider the trade-off between exploitation (choosing the “luckier” machines) and exploration (checking which machines are the “lucky” ones).

Our key idea is to draw an analogy between our task and the multi-armed bandit problem. Since we hope to explore the performance of existing algorithms while exploiting the one with the best performance, it is intuitive to make each algorithm represent one bandit machine in the multi-armed bandit problem. The analogy faces two immediate difficulties: how to identify an appropriate multi-armed bandit method to solve the problem, and how to design a reward scheme that connects the goal of active learning to the goal of the multi-armed bandit problem.

Next, we resolve the two difficulties and explain our proposed approach in detail. Then, we introduce one state-of-the-art approach that not only works for the task but also is based on the multi-armed bandit problem, and discuss our key differences to the state-of-the-art approach.

3.1 Choice of Multi-armed Bandit Method

Our first task is to identify an appropriate multi-armed bandit method for ALBL. We solve the task by looking at the characteristics of our assumed rewards, which are associated with the learning performance. First, it is intuitive that the rewards are not independent random variables across the iterations, because the learning performance generally grows

as D_l becomes larger. Second, the contributions to the learning performance can be time-varying because different algorithms may perform differently in different iterations (Donmez, Carbonell, and Bennett 2007). Thus, making statistical assumptions about the rewards may be difficult.

The scenario above matches the so-called adversarial setting in the multi-armed bandit problem (Auer et al. 2002). One state-of-the-art method that comes with a strong theoretical guarantee for the adversarial setting is called EXP4.P (Beygelzimer et al. 2011), which is an improvement on an earlier EXP4 (Auer et al. 2002) method. We thus consider EXP4.P as the core solver for our proposed ALBL approach. Both EXP4 and EXP4.P define the concept of experts, which can be viewed as soft mixtures of active learning algorithms in our analogy. For simplicity, in this paper, we only consider special experts that correspond to single algorithms instead of soft mixtures.

Next, let us look at the skeleton of ALBL with EXP4.P as the core solver. EXP4.P adaptively maintains a weight vector $\mathbf{w}(t)$ in iteration t , where the k -th component $w_k(t)$ is the non-negative weight of the k -th expert that simply corresponds to the k -th active learning algorithm. The weight vector $\mathbf{w}(t)$ is then scaled to a probability vector $\mathbf{p}(t) \in [p_{\min}, 1]^K$ with some parameter $p_{\min} > 0$. EXP4.P randomly chooses an expert (active learning algorithm in ALBL) based on $\mathbf{p}(t)$, and obtains the reward r of the choice.

Without loss of generality, assuming that the k -th algorithm a_k is chosen by EXP4.P in ALBL. Then, the query request of a_k should be followed to query from D_u . To accommodate the possibility that a_k would want to make a probabilistic query, we introduce the query vector $\psi^k(t) \in [0, 1]^{n_u}$, where its j -th component $\psi_j^k(t)$ indicates the preference of the k -th algorithm on querying the label of $x_j \in D_u$ in iteration t . The query vector should represent a probability distribution of querying from D_u ; that is, $\sum_{j=1}^{n_u} \psi_j^k(t) = 1$. Deterministic active learning algorithms could simply return a degenerate query vector that contains a single 1 on its most preferred instance, and 0 elsewhere.

There are two probabilistic decision steps above. First, EXP4.P uses $\mathbf{p}(t)$ to choose an active learning algorithm, and then, ALBL takes $\psi^k(t)$ to query the label of some $x_* \in D_u$. The two steps can be combined to directly sample $x_* \in D_u$ based on $q_j(t) = \sum_{k=1}^K p_k(t) \psi_j^k(t)$, the probability of querying the j -th instance in the t -th iteration.

Note that different algorithms a_k may actually suggest querying the same instance. Thus, following one algorithm in ALBL is virtually akin to following the other algorithms that make the same suggestion. In our analogy, the situation would correspond to getting the rewards from multiple bandit machines at the same time, which is something that EXP4.P does not consider. Thus, the original EXP4.P only updates the $w_k(t)$ on the chosen expert k with a re-scaled reward $\frac{r}{p_k(t)}$. Considering the special situation above, we

modify EXP4.P and take $\frac{r\psi_*^k(t)}{q_*(t)}$ to update the $w_k(t)$ on all the k algorithms that make the same suggestion on querying x_* . When only one algorithm suggests x_* , our update formula is equivalent to the one in EXP4.P.

3.2 Choice of Reward Function

After modifying EXP4 as the core solver within the proposed ALBL approach, the remaining task is to design a proper reward function. The ideal reward function shall be the test accuracy of f_t , because the cumulative reward that EXP4 targets at would then correspond to the average test accuracy achieved during the iterations of active learning. However, the test accuracy is impossible to obtain in the real world because a test set is generally not available for active learning due to the costliness of labeling.

Another possibility is the training accuracy of f_t . However, training accuracy may not be the best choice for two reasons. First, it suffers from the inevitable training bias when selecting the best f_t based on the labeled data. Second, it suffers from the sampling bias when using active learning to strategically query the unlabeled instances.

Because ALBL samples from the unlabeled pool probabilistically based on the values of $q_j(t)$, it is actually possible to correct the sampling bias using those values. One correction technique, called importance weighting, was originally designed for stream-based active learning (Beygelzimer, Dasgupta, and Langford 2009). The technique is also utilized in a pool-based active learning algorithm (Ganti and Gray 2012) to select a proper f_t . Here, we extend the technique for a different purpose: providing a proper reward function, called IMPORTANCE-WEIGHTED-ACCURACY, for the modified EXP4 within ALBL. In particular, we apply established results for estimating the expected loss with importance weighting (Ganti and Gray 2012) on the 0/1 loss, which is simply the negative accuracy.

To understand the key idea within the IMPORTANCE-WEIGHTED-ACCURACY technique, let us first assume that the data pool D is fully labeled and each example in D is generated i.i.d. from some distribution that will also be used for testing. Then, it is well-known that for a fixed classifier f , the average accuracy $\frac{1}{n} \sum_{i=1}^n \mathbb{I}[y_i = f(x_i)]$ on D is an unbiased estimator of the test accuracy of f , where i is used to index instances in the entire pool.

The average accuracy requires all examples in D to be labeled. The IMPORTANCE-WEIGHTED-ACCURACY technique utilizes sampling to form another unbiased estimator that does not need all the y_i (Ganti and Gray 2012). Attach a probability value $q_i > 0$ for each example $(x_i, y_i) \in D$, take those values to sample one (x_*, y_*) , and use binary random variables $s_i \in \{0, 1\}$ to denote the outcome of the sampling. Then, for each (x_i, y_i) , let $c_i = \mathbb{I}[y_i = f(x_i)]$, and the expected value of $s_i \frac{c_i}{q_i}$ over the sampling process is simply c_i . Thus, $\frac{1}{n} \sum_{i=1}^n s_i \frac{c_i}{q_i} = \frac{1}{n} \frac{c_*}{q_*}$ is also an unbiased estimator of the test accuracy of f . That is, the accuracy c_* of the sampled example can be re-weighted by $\frac{1}{nq_*}$ to form a simple unbiased estimator of the test accuracy of f .

Recall that the proposed ALBL effectively takes $q_j(t)$ to sample one instance from D_u , which does not fully match the discussions above. In particular, sampling from only D_u means instances $(x_i, y_i) \in D_l$ are attached with $q_i = 0$. Thus, Ganti and Gray (2012) propose to allow re-querying labeled examples with a non-zero probability. Similarly, we design ALBL by incorporating a special algorithm (bandit

machine) RANDOM that randomly selects one instance from the *entire* data pool. The design strengthens ALBL in two aspects. First, no modification of other active learning algorithms is needed, making it possible to re-use existing algorithms easily. Second, RANDOM serves as naïve sampling strategy that is sometimes competitive (see Section 4) to active learning algorithms. Incorporating RANDOM provides ALBL with an alternative when other human-designed strategies fail. Note that RANDOM is sufficient for serving the need, but not necessary. One interesting future direction is to study other possibilities than RANDOM.

Because instances in D_l can now be re-queried, we now assume $\psi^k(t)$ to be of length n rather than n_u . Assume that instance i_t is queried in iteration t of ALBL, and let $W_t = (q_{i_t}(t))^{-1}$. For any fixed classifier f , define the IMPORTANCE-WEIGHTED-ACCURACY (IW-ACC) after τ iterations to be

$$\text{IW-ACC}(f, \tau) = \frac{1}{nT} \sum_{t=1}^{\tau} W_t \mathbb{1}[y_{i_t} = f(x_{i_t})].$$

Then, the following theorem shows that $\text{IW-ACC}(f, \tau)$ is an unbiased estimator of the test accuracy of f if (x_i, y_i) are i.i.d.-generated from the test distribution.

Theorem 1. *For any τ , $\mathbb{E}[\text{IW-ACC}(f, \tau)] = \frac{1}{n} \sum_{i=1}^n \mathbb{1}[y_i = f(x_i)]$, where the expectation is taken over the randomness of sampling independently in iteration 1, 2, ..., τ .*

Proof. The theorem can be proved by averaging the simple estimator obtained from each iteration, and is a special case of Theorem 1 made by Ganti and Gray (2012). \square

The proposed ALBL simply takes $\text{IW-ACC}(f_t, t)$ as the reward in the t -th iteration to evaluate how much the chosen algorithm a_k helps getting a better f_t . Combining the ideas of modifying EXP4.P, incorporating RANDOM, and taking IW-ACC as the reward, we list the full ALBL approach in Algorithm 1. The probabilistic nature of EXP4.P and the use of RANDOM allows the reward to be an unbiased estimator of the test performance, making ALBL truly *by learning*—connected with the performance measure of interest.

3.3 A Related Blending Approach

COMB (Baram, El-Yaniv, and Luz 2004) is a state-of-the-art adaptive blending approach that applies EXP4 to solve the task of adaptively choosing active learning. At first glance, ALBL seems similar to COMB in applying multi-armed bandit methods for the task. A closer look reveals two key differences, as discussed below.

Whereas ALBL takes the candidate active learning algorithms as the bandit machines, COMB draws an analogy that takes the unlabeled examples as the bandit machines instead. As a consequence, COMB has to choose from a large and varying numbers of bandit machines, and also has to cope with the restriction that each bandit machine can only be pulled once. The properties contradict the original settings of EXP4, which considers a moderate and fixed number of bandit machines, each of which can be pulled for multiple times. Thus, COMB relies on quite a few heuristic modifications of the original EXP4 to work properly.

Further, the COMB approach takes a human-designed criterion called CLASSIFICATION ENTROPY MAXIMIZATION (CEM) as the reward. CEM is defined as the entropy of f_t -predicted labels in D_u . While some empirical evidence shows that CEM can sometimes match the test accuracy, there is no formal connection between CEM and the test accuracy. The ALBL approach, on the other hand, takes an unbiased estimator for the test accuracy as the reward, which is directly related to the performance measure of interest and can hence be called *by learning*.

4 Experiments

We incorporate four algorithms within our proposed ALBL approach. The first algorithm is called RANDOM, which was previously introduced in Section 3.2. The other three are existing active learning algorithms introduced in Section 2: UNCERTAIN (Tong and Koller 2002), PSDS (Donmez and Carbonell 2008), and QUIRE (Huang, Jin, and Zhou 2010). Each of the algorithms covers a different design philosophy used by humans in active learning. In addition, as we shall show next, each algorithm performs strongly on some data sets but can be relatively worse on the others. That is, no algorithm is an overall winner, which suggests that choosing and blending the algorithms subject to different data sets is important. We take SVM (Vapnik 1998) as the underlying classifier and use LibSVM (Chang and Lin 2011) with all the default parameters to train the classifier.

We first compare ALBL with the four algorithms it incorporates. Next, we examine whether ALBL can compete with a naïve blending approach that uses a fixed ratio. Finally, we demonstrate the benefits of using the unbiased estimator in ALBL by comparing it with two related approaches: the state-of-the-art blending approach COMB (Baram, El-Yaniv, and Luz 2004) and a modified version of ALBL called ALBL-TRAIN that takes the training accuracy as the reward.

We take six real-world data sets, *liver*, *sonar*, *vehicle*, *breast*, *diabetes*, *heart*) from the UCI Repository (Bache and Lichman 2013). For each data set, we reserve 80% of the instances as the training set, and retain the other 20% as the test set to evaluate the test accuracy (see Section 2). Then, from the training set, we randomly select one instance of each class as the initial labeled pool D_l . Each experiment is averaged over ten runs.

4.1 ALBL versus Underlying Algorithms

Figure 1 shows comparison between ALBL with the underlying algorithms on test accuracy, which confirms our statement that no single algorithm can provide consistently superior performance. For instance, QUIRE (the purple curve) performs strongly on *diabetes*; PSDS (the blue curve) is promising on *sonar*; UNCERTAIN (the green curve) dominates on *vehicle* and *liver*.

From Figure 1, we see that ALBL is usually close to the best curves of the four underlying algorithms, except in *liver* which harder to learn (accuracy close to 50%). In Table 1, we further compare ALBL with the four algorithms with a two-sample t-test at 95% significance level when querying different percentage of instances from the unlabeled pool.

Algorithm 1 ACTIVE LEARNING BY LEARNING**Input:** $D = (D_u, D_l)$: data pool; T : query budget; $A = \{a_1, \dots, a_K\}$: active learning algorithms, one of which is RANDOM**Initialize:**

- 1: set $t = 1$, $budget_used = 0$
- 2: **while** $budget_used < T$ **do**
- 3: run EXP4.P for one iteration and obtain the choice vector $\mathbf{p}(t)$
- 4: for all x_i in D , calculate $q_i(t)$ using $\mathbf{p}(t)$ from EXP4.P and $\psi^k(t)$ from all the a_k 's
- 5: sample an instance x_{i_t} based on $q_i(t)$, and record $W_t = (q_{i_t}(t))^{-1}$
- 6: **if** $x_{i_t} \in D_u$ (i.e., has not been queried) **then**
- 7: query y_{i_t} , move (x_{i_t}, y_{i_t}) from D_u to D_l , and train a new classifier f_t with the new D_l
- 8: $budget_used = budget_used + 1$
- 9: **else**
- 10: $f_t = f_{t-1}$ because D_l is not changed
- 11: **end if**
- 12: calculate reward $r = IW-ACC(f_t, t)$
- 13: feed r to a modified EXP4.P that updates the weights of all the algorithms (experts) that suggest x_{i_t}
- 14: $t = t + 1$
- 15: **end while**

Table 1: Win/tie/loss counts of ALBL versus underlying algorithms based on a two-sample t-test

rank	percentage of queried instances							total
	5%	10%	15%	20%	30%	40%	50%	
1st	1/5/0	1/5/0	1/4/1	0/5/1	0/6/0	0/6/0	0/6/0	3/37/2
2nd	2/4/0	1/5/0	1/5/0	0/5/1	1/5/0	1/5/0	0/6/0	6/35/1
3rd	2/4/0	1/5/0	1/5/0	1/5/0	2/4/0	1/5/0	2/4/0	10/32/0
4th	4/2/0	4/2/0	3/3/0	2/4/0	4/2/0	4/2/0	4/2/0	25/17/0
total	9/15/0	7/17/0	6/17/1	3/19/2	7/17/0	6/18/0	6/18/0	44/121/3

The four algorithms are ranked in terms of their mean accuracy on the data set under the particular percentage of queried instances. The results demonstrate that ALBL often yields comparable performance with the better of the four algorithms and can sometimes perform even better. Note that the comparable performance to the better algorithms readily demonstrates that ALBL can solve the challenging task of making reasonable choices from several different algorithms in a data-dependent and adaptive manner.

4.2 ALBL versus Fixed Combination

One question that may be asked is whether it is necessary to use dynamic sampling weights, $\mathbf{p}(t)$, such as used by ALBL. To answer the question, we compare ALBL with a naïve blending approach, named FIXEDCOMB, that takes fixed sampling weights. The performance of ALBL was compared to that of FIXEDCOMB when incorporating two active learning algorithms, one of which reaches the best performance and the other reaches the worst performance on each data set. Further, we consider sampling weight ratios: 10:0, 8:2, 6:4, 5:5, 4:6, 2:8, and 0:10 in FIXEDCOMB. Owing to space limitations and readability, only selected curves for two data sets, *breast* and *diabetes* are shown. The two underlying algorithms are UNCERTAIN and QUIRE for *breast*, and PSDS and UNCERTAIN for *diabetes*. Experiments on other data sets have shown similar results.

Figure 2 reveals two drawbacks of FIXEDCOMB. First, similar to the difficulty of choosing the underlying al-

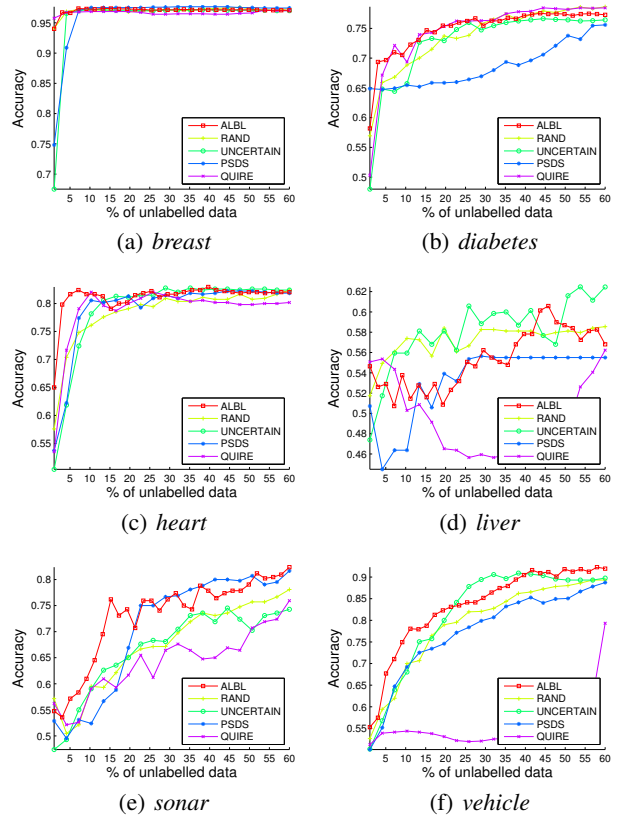


Figure 1: Test accuracy of ALBL and underlying algorithms

gorithms, deciding the best weight ratio beforehand is a very challenging endeavor. For instance, on *breast*, the best weight ratio is 6:4 for querying 10% of the instances, whereas on *diabetes*, the best weight ratio is 4:6. The second drawback is that FIXEDCOMB cannot capture the time-varying behavior of the underlying algorithms. For instance,

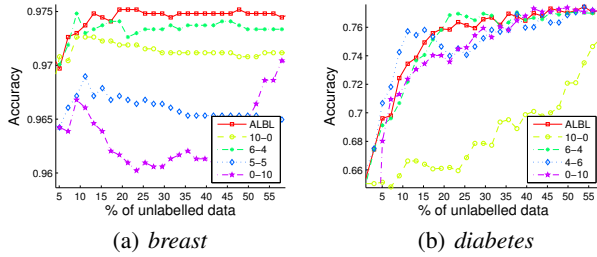


Figure 2: Test accuracy of ALBL versus FIXEDCOMB

on *breast*, weight ratio 0:10 is the least favored in the beginning, but surpasses the weight ratio 5:5 in the end. ALBL resolves the drawbacks by dynamically adjusting the weights towards a better ratio based on the estimated learning performance of the algorithms. Thus, ALBL achieves competitive or even superior performance to the best ratio in the FIXED-COMB family. The results justify adoption of EXP4.P to dynamically decide the sampling weights.

4.3 ALBL versus Two Adaptive Approach

Next, we compare ALBL with COMB, another blending approach based on a modified EXP4 algorithm and a human-designed reward function called CEM. To make the comparison more fair, we show the results on a sibling version of COMB based on EXP4.P (as in ALBL) coupled with CEM. Some side experiments show that the sibling version performs very similarly to the original COMB approach. In addition, as a baseline adaptive active learning approach, we also include in the comparison a modified version of ALBL, called ALBL-TRAIN, that takes the training accuracy of the classifier as the reward.

Figure 3 shows the comparison between ALBL, COMB, and ALBL-TRAIN on test accuracy. The three algorithms sometimes reach comparable performance, such as on *diabetes*. On most of the other data sets, ALBL achieves superior performance to those of the COMB and ALBL-TRAIN. We further analyze the superior performance by evaluating IW-ACC, CEM, the training accuracy, and the true test accuracy at each iteration of ALBL, and depict two representative results in Figure 4. For *diabetes*, all of the three estimators are quite close in tracing the test accuracy in Figure 4(a), which explains the comparable performance in Figure 3(b). On the other hand, for *heart* in Figure 4(b), CEM is rather inaccurate in estimating the test accuracy, and the training accuracy overestimates the test accuracy when there are only a few instances. The inaccurate estimations results in worse performance of COMB and ALBL-TRAIN than ALBL. The results confirm the benefits of using a sound estimator of the learning performance (IW-ACC) as the reward instead of a human-designed criterion such as CEM.

5 Conclusion

We propose a pool-based active learning approach ALBL that allows the machines to conduct active learning by learning. ALBL adaptively and intelligently chooses among vari-

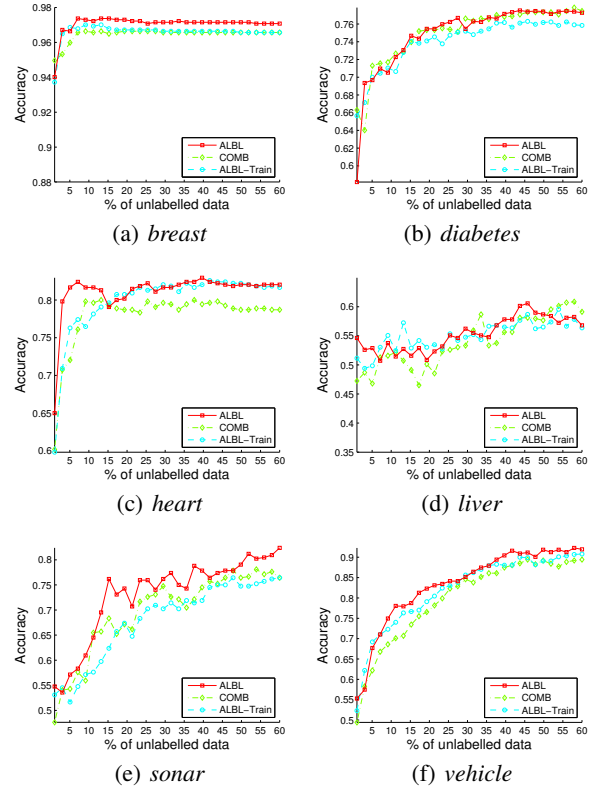


Figure 3: Test accuracy of ALBL, COMB, and ALBL-TRAIN

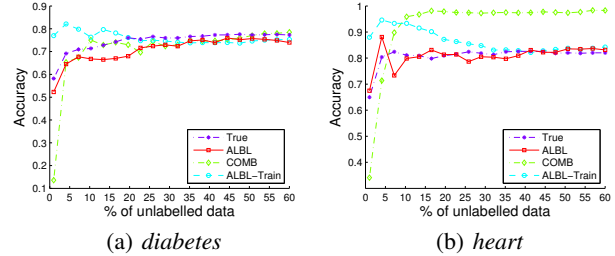


Figure 4: Different estimations of true test accuracy

ous existing active learning strategies by using their learning performance as feedback. We utilize the famous EXP4.P algorithm from the multi-armed bandit problem for the adaptive choice, and estimate the learning performances with IMPORTANCE-WEIGHTED-ACCURACY. Extensive empirical studies lead to three key findings. First, ALBL is effective in making intelligent choices, and is often comparable to or even superior to the best of the existing strategies. Second, ALBL is effective in making adaptive choices, and is often superior to naïve blending approaches that randomly choose the strategies based on a fixed ratio. Third, ALBL is effective in utilizing the learning performance, and is often superior to the human-criterion-based blending approach COMB. These findings indicate that the proposed ALBL is a favorable and promising approach in practice.

6 Acknowledgments

We thank the anonymous reviewers and the members of the NTU Computational Learning Lab for valuable suggestions. This work is partially supported by the Ministry of Science and Technology of Taiwan via the grant MOST 103-2221-E-002-148-MY3.

References

- Auer, P.; Cesa-Bianchi, N.; Freund, Y.; and Schapire, R. E. 2002. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing* 32(1):48–77.
- Bache, K., and Lichman, M. 2013. UCI machine learning repository.
- Baram, Y.; El-Yaniv, R.; and Luz, K. 2004. Online choice of active learning algorithms. *The Journal of Machine Learning Research* 5:255–291.
- Beygelzimer, A.; Langford, J.; Li, L.; Reyzin, L.; and Schapire, R. E. 2011. Contextual bandit algorithms with supervised learning guarantees. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, 19–26.
- Beygelzimer, A.; Dasgupta, S.; and Langford, J. 2009. Importance weighted active learning. In *Proceedings of the 26th International Conference on Machine Learning*, 49–56.
- Chang, C.-C., and Lin, C.-J. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2:27:1–27:27.
- Chu, W.; Zinkevich, M.; Li, L.; Thomas, A.; and Tseng, B. 2011. Unbiased online active learning in data streams. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 195–203.
- Cohn, D.; Atlas, L.; and Ladner, R. 1994. Improving generalization with active learning. *Machine learning* 15(2):201–221.
- Dasgupta, S., and Hsu, D. 2008. Hierarchical sampling for active learning. In *Proceedings of the 25th International Conference on Machine Learning*, 208–215.
- Donmez, P., and Carbonell, J. G. 2008. Paired-sampling in density-sensitive active learning. In *Proceedings of the International Symposium on Artificial Intelligence and Mathematics*.
- Donmez, P.; Carbonell, J. G.; and Bennett, P. N. 2007. Dual strategy active learning. In *Proceedings of the 18th European Conference on Machine Learning*, 116–127.
- Ganti, R., and Gray, A. 2012. UPAL: Unbiased pool based active learning. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*, 422–431.
- Huang, S.-J.; Jin, R.; and Zhou, Z.-H. 2010. Active learning by querying informative and representative examples. In *Advances in Neural Information Processing Systems*, volume 23, 892–900.
- Jacobs, R. A.; Jordan, M. I.; Nowlan, S. J.; and Hinton, G. E. 1991. Adaptive mixtures of local experts. *Neural computation* 3(1):79–87.
- Lewis, D. D., and Gale, W. A. 1994. A sequential algorithm for training text classifiers. In *Proceedings of the 17th ACM International Conference on Research and Development in Information Retrieval*, 3–12.
- Nguyen, H. T., and Smeulders, A. 2004. Active learning using pre-clustering. In *Proceedings of the 21st International Conference on Machine Learning*, 623–630.
- Robbins, H. 1985. Some aspects of the sequential design of experiments. In *Herbert Robbins Selected Papers*. Springer. 169–177.
- Settles, B. 2010. Active learning literature survey. Technical report, University of Wisconsin, Madison.
- Tong, S., and Koller, D. 2002. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research* 2:45–66.
- Vapnik, V. 1998. *Statistical learning theory*. Wiley.
- Vermorel, J., and Mohri, M. 2005. Multi-armed bandit algorithms and empirical evaluation. In *Proceedings of the 16th European Conference on Machine Learning*, 437–448.