2010 IEEE International Conference on Data Mining

# Active Learning from Multiple Noisy Labelers with Varied Costs

Yaling Zheng, Stephen Scott
*Department of Computer Science & Engineering*
*University of Nebraska-Lincoln*
*Lincoln, Nebraska*
*Email: {yzheng, sscott}@cse.unl.edu*

Kun Deng
*Department of Statistics*
*University of Michigan*
*Ann Arbor, MI*
*Email: kundeng@umich.edu*

***Abstract*—In active learning, where a learning algorithm has to purchase the labels of its training examples, it is often assumed that there is only one labeler available to label examples, and that this labeler is noise-free. In reality, it is possible that there are multiple labelers available (such as human labelers in the online annotation tool Amazon Mechanical Turk) and that each such labeler has a different cost and accuracy. We address the active learning problem with multiple labelers where each labeler has a different (known) cost and a different (unknown) accuracy. Our approach uses the idea of *adjusted cost*, which allows labelers with different costs and accuracies to be directly compared. This allows our algorithm to find low-cost combinations of labelers that result in high-accuracy labelings of instances. Our algorithm further reduces costs by pruning under-performing labelers from the set under consideration, and by halting the process of estimating the accuracy of the labelers as early as it can. We found that our algorithm often outperforms, and is always competitive with, other algorithms in the literature.**

***Keywords*-active learning; multiple labelers; noisy labelers; algorithms; adjusted cost**

## I. Introduction

In active learning, it is assumed that unlabeled data is easy to obtain but labels are expensive. For example, when building a speech recognizer, it is easy to get raw speech samples, but labeling the samples is a tedious process in which a human must examine the speech signal and carefully segment it into phonemes. Therefore, a subset of instances is carefully chosen to be labeled by a labeler, and the remaining instances remain unlabeled. The goal is to learn a classifier by labeling as few instances as possible by actively selecting the instances to be labeled as learning proceeds. Many results in active learning focus on choosing the instances for labeling and assume that the labeling is handled by a single, noise-free labeler [1], [2], [3]. However, it is possible that there is no perfect labeler and that instead multiple, noisy labelers are available. For example, consider the aforementioned speech recognition application. It is difficult to guarantee 100% accuracy from a human labeler, due to the wide variations in how we interpret speech signals. However, one may easily have access to multiple human labelers, each with a different cost.

Another example of multiple noisy labelers is the online annotation tool Amazon Mechanical Turk (AMT) [4]. AMT requesters submit *human intelligence tasks* (HITs) to the site. A HIT is a task that is simple for humans but may be difficult to automate. Examples include: draw an outline around a human in a given picture, solve an optical character recognition task, etc. For their efforts, AMT workers are paid a fee, based on how many data items they label. Certainly, in the AMT model, one cannot guarantee noise-free labels from any human labelers, and in fact, depending on the task, noise rates may vary considerably. Finally, while requesters who submit the HITs set the pay rates, it is not conceptually difficult to extend the AMT idea to that of a marketplace in which the workers (labelers) would bid on jobs, using strong on-line ratings to justify higher costs. Such a marketplace would have variance both in cost and in quality of labelers.

We refer to this model as *active learning with multiple noisy labelers,* where the goal is to learn a hypothesis that generalizes well while spending as little as possible on queries to labelers. We will assume that the label given by labeler $o_i$ for a particular instance will be the true label, corrupted by noise with probability $\eta_i < 1/2$, which is $o_i$'s noise rate. We assume that the noise model is an i.i.d. process, where the independence is across labelers (one labeler's label of instance $x$ is not influenced by the label issued by any other labeler for $x$) and across instances (a labeler's label for instance $x$ is independent of the label it issued for instance $x'$, which it saw earlier). Finally, we assume that each labeler's noise model is *persistent* in that if labeler $o_i$ labels instance $x$ as class $y$ at some point in the run of the algorithm, then it will always answer $y$ as its label for $x$. Thus, there is no more information to be had by asking a labeler to label the same instance multiple times. Note that there is no hard budget in our problem definition. Instead, the goal in our model is to simply spend as little as possible while generalizing well.

Sheng et al. [5] show that when labelers provide noisy labels under the persistent noise model, one can still estimate well the true labels of instances by requesting labels on a single instance from multiple, independent labelers. The idea is that, if all labeler responses are independent from each other and all noise rates are $< 1/2$, then a majority vote from an appropriate subset of the labelers can be very effective. (We extend this concept into part of our

1550-4786/10 $26.00 © 2010 IEEE
DOI 10.1109/ICDM.2010.147

algorithm: the notion of combined accuracy of a group of labelers.) However, in their work they assume that all the labelers have the same costs and noise rates. Donmez et al. [6] relaxed this assumption and assumed that labelers can have different, unknown accuracies. They proposed a method called IEThresh for active learning with multiple noisy labelers with different accuracies.

Briefly, IEThresh works as follows: First, the algorithm chooses an instance for labeling. Then it compares the relative performance of the labelers so far (via interval estimation learning [7]) to choose what it believes to be the most accurate labelers to label the chosen instance. Next, it estimates the "ground truth" (true label) of the instance by a majority vote of these chosen labelers. At that point it updates its classifier based on the newly labeled training instance. Finally, it updates its estimate of each labeler's accuracy, and repeats the process.

Contrasting IEThresh to the naïve approach of simply having every labeler label every instance and then taking a majority vote (what is sometimes called "repeated labeling" in the literature and what we refer to as "Repeated" in our experimental section), the advantage of IEThresh is obvious. The execution of IEThresh excludes labelers believed to be inferior, which saves total cost over Repeated. However, while effective, IEThresh does not consider the possibility that one can further save labeling cost when many accurate labelers exist by further pruning the set of labelers used. Our new algorithm IEAdjCost overcomes this shortcoming.

The intuition behind IEAdjCost is that we "normalize" the accuracies of labelers with respect to their costs, allowing for direct comparison between labelers. Then, rather than simply identifying the most accurate labelers, our algorithm instead seeks a subset of labelers that, when combined, achieves the desired level of accuracy for the least cost.

IEAdjCost works in two phases. In Phase 1, it uses higher accuracy labelers to predict the ground truth by majority vote, and it stops estimating labeler accuracy when it has a sufficiently large set of labelers with good accuracy estimates. In Phase 2, a heuristic finds a subset of the labelers with good accuracy estimates that, when used together, has high accuracy and low cost. This subset is what is used from that point forward for labeling requests.

Thus, in Phase 1, there exist both exploration (estimating the accuracies of labelers) and exploitation (estimating ground truth by using only subset of labelers believed to be most accurate). In Phase 2, there is no exploration and IEAdjCost chooses a final set of labelers that, when combined, is believed to have high accuracy and low cost. (In Phase 2, we stop refining the accuracy estimates of the labelers since by definition of the algorithm we already have sufficiently good estimates for our purposes and further refinement would be an unnecessary expense.)

Another drawback of Donmez et al.'s IEThresh is that it ignores the possibility that labelers can have different costs. In reality, there could be high variance in the costs of available labelers, and it may be the case that the costs may not even correlate with accuracy (expensive labelers may not necessarily be highly accurate). But even in cases where cost correlates with accuracy, it is possible that multiple inexpensive labelers, when used together in a voting scheme, may have a greater accuracy than one highly expensive labeler. For these reasons, active learning with multiple labelers, when both cost and accuracy vary, is a more complex problem than the case of identical costs, as addressed by IEThresh. Thus we introduce the notions of *adjusted cost* and *combined accuracy* that allow us to directly compare labelers with different costs and accuracies. Combined accuracy gives us the probability that a group of independent labelers will make a labeling mistake when the label is found by a majority vote within that group. The adjusted cost of labeler $o_j$ is the cost of $o_j$ multiplied by the number of independent copies[1] of $o_j$ needed to achieve a particular combined accuracy.

In Phase 2, we take a set of labelers who have their accuracies estimated well and select a subset of them with low total cost and high combined accuracy. We propose a heuristic called LabelersByAdjCost to solve this problem. LabelersByAdjCost first sorts the given labelers by their adjusted costs in ascending order, and then grows the final set of labelers in the order of increasing adjusted cost until the combined accuracy of the group exceeds a threshold.

This rest of the paper is organized as follows. Section II describes related work. Section III presents our new algorithm IEAdjCost. Our experiments are summarized in Section IV, comparing our algorithm IEAdjCost to Repeated and IEThresh on six data sets from UCI [8] and two data sets from AMT [4], [9]. We conclude in Section V.

## II. Related Work

One of the primary objectives of engineering machine learning systems is to reduce the expense of human effort. Ironically, in many applications, significant amounts of human labor and time are still spent in preparing high-quality data in order to build such intelligent systems.

If we treat data preparation as an independent "post-processing" step after the data is initially collected, one way of utilizing incomplete, noisy or erroneous labeling sources without manual intervention is to infer the truth. The EM algorithm of Dempster et al. [10] is a popular procedure for finding maximum likelihood estimates of parameters where the model depends on unobserved latent variables. Dawid and Skene [11] proposed using EM to improve the quality of simple majority voting by experts in the biostatistics community. They showed that by using EM, one can better estimate the ground truth and therefore the accuracies of each expert.

[1]This is purely hypothetical. In the persistent noise model, using the same labeler repeatedly on a single instance yields no new information.

Smyth et al. [12] took advantage of the EM algorithm for the application of a large-scale image analysis problem. They studied the problem of inferring volcano types based on labelings from multiple labelers. The EM algorithm was applied to estimate the conditional distribution of a volcano's type given noisy labels. This information was then used as the ground truth to learn a classifier.

Sheng et al. [5] studied the problem of acquisition of labels from multiple imperfect labelers, and one of their conclusions was that repeated labeling (getting an instance's label once from each of several labelers) is often preferable to single labeling (getting an instance's label once from a single labeler), even when labels are not particularly cheap. In another study in the natural language processing domain, Snow et al. [9] concluded that multiple cheap labelers can be preferable to a single expert. They also proposed a bias correction technique by checking the labelers' performance on the "gold standard" (true labels) in order to fine tune each labeler's relative weight. By doing experiments on five tasks from the Amazon Mechanical Turk system [4], they showed that significant improvements in annotation quality can be achieved at a fraction of the usual expense. In the medical domain, where an expert's performance is usually described by the trade-off of sensitivity and specificity, another study by Raykar et al. [13] showed a significant advantage of combining the opinions of multiple experts when training a logistic regression classifier.

Donmez and Garbonell [14] provided a decision-theoretic framework to select labelers and instances in an active learning setting. However, their algorithms required at least one labeler to be noise-free and able to answer all queries. They studied scenarios where (1) one labeler answers every query and is noise-free and the other is noise-free but declines to answer some queries; (2) both labelers answer all queries, one noise-free and one noisy; and (3) both labelers answer all queries without noise, but one labeler has a fixed cost per query and the other labeler's cost depends on the instance's class posterior probability.

Finally, Donmez et al. [6] extended Donmez and Garbonell's work to remove the limitations that there are only two labelers and that one labeler must be perfect. They proposed a method called IEThresh that takes multiple noisy labelers and chooses those labelers with high accuracy to both save cost and improve accuracy. They studied active learning in the case of multiple noisy labelers with identical costs. Their method assumes that each labeler's accuracy is strictly better than random guessing, which implies that majority voting will, in expectation, yield correct labels. They reduced the cost of labeling by filtering out labelers that are not as good as the others.

Donmez et al.'s work can be improved in two ways. First, after estimating the accuracies of the labelers, one doesn't need to select all of those that are highly accurate. Instead, one only needs to choose enough such labelers to achieve the desired rate of combined accuracy when using majority voting. Second, instead of assuming that all labelers have identical costs, labelers with different costs and accuracies should be directly compared, and the goal should be to choose a set of labelers whose combined accuracy is high but total cost is low. Our algorithm IEAdjCost makes both of these improvements. IEAdjCost will avoid selecting labelers, even if they are accurate, if they are expensive and not needed to achieve the required combined accuracy.

## III. Algorithm IEAdjCost

In this section, Section III-A introduces notation. Section III-B summarizes the principle of uncertainty sampling [15], [1] used in IEAdjCost to choose instances to label. In Section III-C, we review interval estimation [7], which is what we use to estimate the accuracies of the labelers, and we describe the various subsets of labelers that we use in our algorithm and their roles. Section III-D defines our concepts of combined accuracy and adjusted cost, which are used by our heuristic LabelersByAdjCost (Section III-E) that finds a final set of labelers with high accuracy and low cost, to be used in Phase 2. Finally, Section III-F puts the pieces together into the full algorithm IEAdjCost.

### A. Notation

We consider the problem of supervised classification for which a training set $\{(x_1, y_1), \cdots, (x_n, y_n)\}$ is used to induce a classifier, where $\{x_1, \cdots, x_n\} = X$ is the set of instances and $\{y_1, \cdots, y_n\} = Y$ is the set of labels. We assume that the instances $x_i$ are given, while the labels $y_i$ are only obtained by purchasing them from one or more labelers from the set $O = \{o_1, \ldots, o_N\}$. (For convenience, we will partition $X$ into $U$ and $L$, where $U$ is the set of instances for which a label has not yet been purchased, and $L$ is the set of labeled instances.[2]) The cost of purchasing a label from labeler $o_i \in O$ is $c_i$ and the label returned by $o_i$ is correct with probability $a_i = 1 - \eta_i$ (labeler $o_i$'s *accuracy*). We denote by $\hat{a}_i$ our algorithm's estimate of this accuracy.

### B. Selecting Instances

In each iteration of our algorithm, the first step is to choose an unlabeled instance $x^* \in U$ for labeling. Our algorithm chooses the instance to label via uncertainty sampling [15], [1], in which one chooses the unlabeled instance for which the current hypothesis is least certain in its classification. Specifically, if $\hat{P}(y \mid x)$ is the current hypothesis's estimate of the probability that instance $x$ has label $y$, then the instance selected for labeling is

$$x^* = \operatorname*{argmax}_{x \in U} \left(1 - \max_{y \in \mathcal{Y}} \hat{P}(y \mid x)\right) \quad . \tag{1}$$

[2]The actual label that goes to $L$ is explained in Algorithm 3.

### C. Estimating Labeler Accuracy

Our algorithm IEAdjCost consists of two phases. Phase 1 is the exploration and exploitation phase and Phase 2 is the pure exploitation phase. In Phase 1, IEAdjCost purchases labels from many labelers (exploration) while simultaneously using the labels offered by a select subset of labelers to train the classifier (exploitation). In Phase 2, IEAdjCost selects a final subset of labelers to label training data.

To estimate the accuracies of the labelers, we adapt the procedure of interval estimation from Kaelbling [7], which was also applied by Donmez et al. [6]. Our algorithm estimates the accuracy $a_i$ of labeler $o_i$ by asking $o_i$ to label several instances from $U$ and giving that labeler a *reward* of 1 for each instance that it labels correctly, and a reward of 0 otherwise. Then our estimate $\hat{a}_i$ of $a_i$ is the sample mean of the rewards that $o_i$ received. We also compute $\hat{s}_i$, the sample standard deviation. We then compute the upper interval ($\text{UI}_i$) and lower interval ($\text{LI}_i$) of an $(\alpha/2)$-level confidence interval centered on $\hat{a}_i$:

$$\text{UI}_i = \hat{a}_i + t_{\alpha/2}^{(n_i-1)} \frac{\hat{s}_i}{\sqrt{n_i}} \quad (2)$$

$$\text{LI}_i = \hat{a}_i - t_{\alpha/2}^{(n_i-1)} \frac{\hat{s}_i}{\sqrt{n_i}} \ , \quad (3)$$

where $n_i$ is the number of rewards (of value 0 or 1) given to labeler $o_i$ (i.e., the number of values used to compute $\hat{a}_i$ and $\hat{s}_i$), and $t_{\alpha/2}^{n_i-1}$ is the critical value for the Student's $t$ distribution with $n_i - 1$ degrees of freedom at the $\alpha/2$ confidence level. In our experiments, we set $\alpha = 0.05$. Since the true labels of instances are unavailable, we estimate the true labels by a majority vote of a select subset of the labelers, which we describe later. To avoid trivialities in the formulas, we initialize each labeler's set of rewards to be $\{0, 1\}$.

The first step of Phase 1 is for IEAdjCost to determine which subset of labelers will have their accuracy estimates refined. IEAdjCost uses a parameter $\lambda$ (in our experiments, we tried $\lambda \in \{0.01, 0.25, 0.5, 0.75, 1\}$) that specifies the fraction of labelers from $O$ that must be explored. Define

$$\text{rank}(o_i) = 1 + |\{o_j \in O : \text{UI}_j > \text{UI}_i\}|$$

as 1 plus the number of labelers in $O$ that have a larger upper interval as labeler $o_i$, i.e. it is $o_i$'s rank in a partial ordering of the labelers based on upper interval. Now we define the set of labelers whose accuracy estimates will be refined:

$$O_\ell = \{o_k \in O : \text{rank}(o_k) \leq \lceil \lambda |O| \rceil\} - O_g \ , \quad (4)$$

where $O_g$ is the set of labelers that have sufficiently good estimates of their accuracies (see below).

To estimate the accuracies of labelers in $O_\ell$, we need an estimate of the correct label of the instance $x^*$. This is computed by taking a majority vote of a separate set of labelers, which we denote $O_r$:

$$O_r = \left\{ o_i : \text{UI}_i \geq \epsilon \max_{o_j \in O} \text{UI}_j \right\} \ , \quad (5)$$

where $\epsilon \in [0, 1]$ is a parameter (in our experiments, we tried $\epsilon \in \{0.7, 0.8, 0.9\}$). Thus in each iteration in Phase 1, IEAdjCost requests labels of the chosen instance from the labelers in $O_\ell \cup O_r$. From the results of the labelings, we can update the rewards for each labeler $o_i \in O_\ell$, and then update $\text{UI}_i$ and $\text{LI}_i$. (Since it doesn't cost any more, we also update UI and LI of labelers in $O_r - O_\ell$.)

Once a labeler's accuracy is well-estimated, we put it in:

$$O_g = \{o_i \in O : \text{UI}_i - \text{LI}_i \leq \delta\} \ . \quad (6)$$

(We tried $\delta \in \{0.2, 0.3, 0.4\}$.) Phase 1 ends when $|O_g| \geq \lceil \lambda |O| \rceil$ and function LabelersByAdjCost($O_g$) returns a non-empty set (Section III-E).

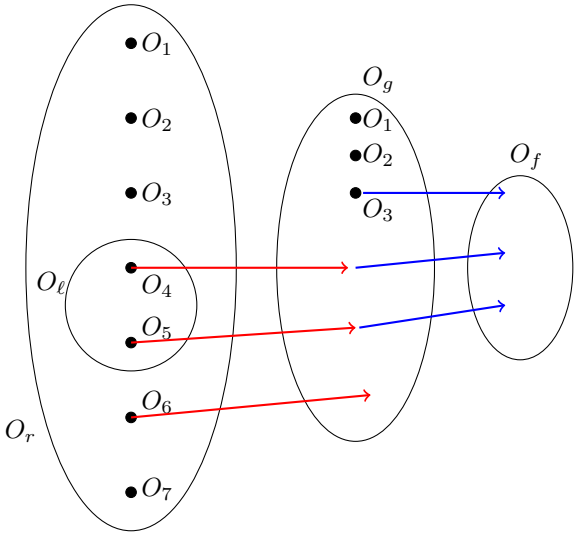


Figure 1. An example of the relationship among $O_\ell$, $O_r$, $O_g$, and $O_f$ in Phase 1.

Figure 1 shows an example of the relationship among $O_\ell$, $O_r$, $O_g$, and $O_f$. To summarize, $O_\ell$ is the set of labelers whose accuracy estimates we are refining, $O_r$ is the set of high-accuracy labelers that are used to predict the ground truth, $O_g$ is the set of labelers whose accuracies are well-estimated, and $O_f$ is the set of final chosen labelers (see Section III-E).

### D. Combined Accuracy and Adjusted Cost

Once $O_g$ is sufficiently large, we switch to Phase 2, the pure exploitation phase. In this phase, a fixed set of labelers is chosen to label all future instances that are added to the labeled training set $L$. (We no longer refine accuracy estimates in Phase 2 because we already have sufficiently good estimates, and continuing to do so only incurs more

cost.) Ideally, we want to choose labelers that together are highly accurate and individually are inexpensive to query. In our process to find such a set of labelers, we use quantities that we call *combined accuracy* and *adjusted cost*.

The notion of combined accuracy of a set $O'$ of labelers is based on the idea that, when taking a majority vote of the labelers in $O'$, we only need at least half of the labelers to correctly label the instance. The combined accuracy of $O' = \{o_1, \ldots, o_k\}$ is the probability that this occurs. Of course, the true accuracies $a_i$ are unavailable, so we compute $\hat{A}$, our estimate of $A$, by substituting our estimated accuracies for the true ones:

$$\hat{A}(o_1, \ldots, o_k) = \sum_{\substack{\{i_1, \cdots, i_{k'}\} \subseteq \{1, \cdots, k\}: \\ k' > k/2}} \left( \prod_{j=1}^{k'} \hat{a}_{i_j} \prod_{\ell \notin \{i_1, \ldots, i_{k'}\}} (1 - \hat{a}_{i_\ell}) \right) \quad . \quad (7)$$

In other words, for every subset of $O'$ that contains a majority of the labelers in $O'$, the products are the probability that that majority subset will correctly label the instance. Summing over all such subsets, we get the probability that *some* majority subset will correctly label the instance.

Using combined accuracy, we can now define the adjusted cost of a labeler, which quantifies its cost/accuracy tradeoff. Central to the concept of the adjusted cost of labeler $o_i$ is its *multiplier* $\beta_i(R)$, which is the number of copies of $o_i$ (in a hypothetical non-persistent noise model) needed to achieve a combined estimated accuracy of at least $R$. I.e., $\beta_i(R)$ is the minimum value of $k$ such that[3]

$$\hat{A}(\overbrace{o_i, \ldots, o_i}^{k}) \geq R \quad .$$

(Since we assume that $R$ is fixed ahead of time as a parameter, for brevity's sake we will omit it from now on and just refer to $o_i$'s multiplier as $\beta_i$.) We can now define the adjusted cost of $o_i$ in terms of its multiplier:

$$\text{AdjustedCost}(o_i) = c_i \beta_i \quad . \quad (8)$$

### *E. Choosing the Final Set of Labelers*

Given a set $O_g \subseteq O$ of distinct labelers whose accuracies are well-estimated, we want to choose a final subset $O_f \subseteq O_g$ to use in Phase 2. The subset we seek is one that achieves a minimum combined accuracy of $R$ whose combined costs are minimized:

$$O_f = \operatorname*{argmin}_{S \subseteq O_g : \hat{A}(S) \geq R} C(S) \quad , \quad (9)$$

where $C(S) = \sum_{o_j \in S} c_j$ is the combined cost of $S$.

To solve this optimization problem, we employ the heuristic LabelersByAdjCost, which first sorts the labelers in $O_g$ in ascending order by their adjusted costs, breaking ties by their estimated accuracies. Then it adds the labelers one at a time to $O_g$ in sorted order, stopping when the combined accuracy of the labelers in $O_g$ exceeds the threshold $R$. Pseudocode for this routine is in Algorithm 1.

[3]In Section III-E, we describe how to compute $\beta_i$.

**Algorithm 1** LabelersByAdjCost($o_1, \ldots, o_N$)

```
Compute adjusted cost for each labeler per Equation (8);
Sort the labelers in ascending order by their adjusted costs (breaking ties by their accuracies in descending order), yielding {o_{r_1}, ..., o_{r_N}} ;
for i from 1 to N do
    Let O' = {o_{r_1}, ..., o_{r_i}} ;
    Sort the labelers of O' by their accuracies in descending order (breaking ties by their costs in ascending order), yielding {o_{z_1}, ..., o_{z_i}} ;
    for j from 1 to i do
        if CombAccuReachesThres(o_{z_1}, ..., o_{z_j}, R) then
            return {o_{z_1}, ..., o_{z_j}} ;
        end if
    end for
end for
return ∅ ;
```

**Algorithm 2** CombAccuReachesThres($o_{z_1}, \ldots, o_{z_j}, R$)

```
m = |{o_{z_1}, ..., o_{z_j}}| ;
if m ≤ N_0 then
    compute the combined accuracy Â of o_{z_1}, ..., o_{z_j} per Equation (7), stop early if Â ≥ R ;
    if Â ≥ R then
        return true ;
    else
        return false ;
    end if
else
    if m ≥ max_{o_i ∈ {o_{z_1}, ..., o_{z_j}}} β_i then
        return true ;
    else
        return false ;
    end if
end if
```

In Algorithm 1, we use combined accuracy to determine each labeler's multiplier to compute its adjusted cost, and to compute the combined accuracy of candidate $O_f$ sets. Below we describe methods we use to perform each task while mitigating the intractability of of Equation (7).

There is a simple means to find the multiplier of a labeler with an accuracy estimate of $\hat{a}$ that leverages the fact that all the accuracies in the hypothetical collection are equal. Since the left-hand side of the following expression is monotonic in $M$, we simply perform a binary search to find the smallest

value of $M$ such that the following expression holds:

$$\sum_{i=0}^{\lceil \frac{M}{2} \rceil - 1} \binom{M}{i} \hat{a}^{M-i}(1-\hat{a})^i \geq R \ .$$

Further, for fixed $R$, the multiplier for accuracy estimate $\hat{a}$ never changes, so we in fact *precompute* the multipliers for each value of $\hat{a} \in \{0.51, 0.52, \ldots, 0.99, 1.00\}$. For example, if $R = 0.95$, then labelers with accuracies $0.8, 0.81, \ldots, 0.99$ will have multipliers $7, 7, 5, 5, 5, 5, 5, 3, 3, 3, 3, 3, 3, 3, 3, 1, 1, 1, 1, 1$.

When building the set $O_f$ in Algorithm 1, iteration $i$ of the loop of Lines 3–11 considers the $i$ labelers in $O_g$ that have the lowest adjusted costs. Call this set $O'$. Since we do not need to know $O'$'s exact combined accuracy, but only whether it exceeds $R$, Line 7 of Algorithm 1 calls CombAccuReachesThres (Algorithm 2) to determine if $O'$ is acceptable to become $O_f$. For increased efficiency, calls to CombAccuReachesThres are on sets that contain the most accurate labelers from $O'$ (Lines 5–10), which will more quickly return a $true$ answer if one is possible[4].

CombAccuReachesThres computes the combined accuracy using Equation (7) if the number of labelers $m \leq N_0$, a parameter that we set to 20 in our experiments, though the computation does stop early once the value exceeds $R$. If instead $m > N_0$, then to improve efficiency, we instead employ the heuristic that returns $true$ if $m$ is at least as large as the largest multiplier in the set of labelers, and $false$ otherwise.

If multipliers are precomputed, then Algorithm 2 has time complexity $O(m + N_0 2^{N_0 - 1})$. In Algorithm 1, sorting the labelers in Line 2 takes $O(N \log N)$ time. In iteration $i$ of the **for** loop, adding labeler $o_{r_i}$ to the already sorted list in Line 5 takes $O(\log i)$ time, and iteration $j$ of the inner **for** loop takes time $O(j + N_0 2^{N_0 - 1})$. Summing over both loops yields a time complexity of $O(N^3 + N^2 N_0 2^{N_0 - 1})$.

### F. Algorithm IEAdjCost

Pseudocode for our main algorithm IEAdjCost is given in Algorithm 3. The parameter $\lambda \in (0, 1]$ used on Lines 5 and 12 specifies the minimum value of $|O_g|/|O|$, i.e. the minimum fraction of original labelers that need to have well-estimated accuracies. Thus $\lambda = 1/|O|$ means that we merely insist on at least one labeler in $O$ to have an accuracy that we estimate well, where $\lambda = 1$ means that we insist that we have good estimates of the accuracies of all labelers.

**Algorithm 3** IEAdjCost

1: $O_g = \emptyset$;
2: **Phase 1:** Initialize rewards for each labeler to $\{0, 1\}$ ;
3: Compute the upper and lower confidence interval for each labeler per Equations (2) and (3) ;
4: Pick the most uncertain unlabeled instance $x^*$ for labeling per Equation (1) ;
5: Build $O_\ell$ per Equation (4) and $O_r$ per Equation (5) ;
6: Purchase labels for $x^*$ from all labelers in $O_\ell \cup O_r$ ;
7: Estimate the ground truth $\bar{y}$ of $x^*$ using majority voting of labelers in $O_r$ ;
8: Update the labeled training data $L = L \cup (x^*, \bar{y})$ and train a new classifier with $L$ ;
9: Update the rewards of all labelers in $O_\ell \cup O_r$ ;
10: Compute the upper and lower confidence interval for each labeler per Equations (2) and (3) ;
11: Compute the set $O_g$ of well-estimated labelers per Equation (6) ;
12: **if** $|O_g| \geq \lceil \lambda |O| \rceil$ and $O_f = \text{LabelersByAdjCost}(O_g) \neq \emptyset$ **then**
13: GOTO Step 17 ;
14: **else**
15: GOTO Step 4 ;
16: **end if**
17: **Phase 2:** Use $O_f$ to label future instances, add them to $L$, and train classifier until training complete ;

## IV. Experiments

We tested IEAdjCost on six data sets from the University of California-Irvine (UCI) repository [8] and two data sets from Amazon Mechanical Turk (AMT) [4], [9]. Section IV-A shows the results of comparing IEAdjCost with IEThresh and Repeated on the UCI data, and Section IV-B presents results for the same algorithms on the AMT data.

### A. Experimental Results on UCI Data Sets

Table I describes the six UCI data sets we tested on: kr-vs-kp, mushroom, car, splice, nursery, and spambase. Data sets car, splice, and nursery are multiclass, but for simplicity we converted them to binary by using partitioning into two classes as done by Rätsch et al. [16]. Data set spambase had continuous attributes, which we discretized using the package `weka.filters.supervised.attribute.Discretize` from Weka [17].

We repeated the following procedure 10 times for each data set. First, we partitioned the data set into 80% training data and 20% test data. Then as an initialization step, at the start of each run we pre-labeled one randomly selected training instance with its correct label, moved that labeled instance from $U$ to $L$, and trained the initial naïve Bayes classifier on $L$. As more points were added to $L$ on Lines 17 and 8 in Algorithm 3, the naïve Bayes classifier was updated.

Each time labeler $o_i$ was asked to label an instance, it returned the true label from the data set with probability $a_i = 1 - \eta_i$ and the opposite label with probability $\eta_i$. We

[4]Note that this efficiency optimization does not necessarily return the subset with smallest adjusted costs since it doesn't return a prefix of $O'$, but we expect the set returned to still have low total cost, since what it returns is a subset of $O'$, which consists of the $i$ labelers with lowest adjusted costs.

set the parameter $R = 0.95$ and ran each algorithm until it purchased 600 instances. This allowed us to witness the asymptotic behavior of each algorithm.

Table I
UCI DATA SETS.

| Data set | Number of Instances | Attributes |
|---|---|---|
| kr-vs-kp | 3196 | 35 |
| mushroom | 8124 | 22 |
| car | 1728 | 4 |
| nursery | 12960 | 8 |
| splice | 3190 | 62 |
| spambase | 4601 | 57 |

*1) Uniform Accuracies:* In our first set of experiments, we repeated the following process five times. We instantiated 50 labelers, the accuracies of which were randomly chosen from (0.5, 1], and the costs of which were randomly chosen from $\{1, \ldots, 30\}$. We tested our new algorithm IEAdjCost with $\lambda \in \{0.02, 0.25, 0.5, 0.75, 1\}$, $\epsilon = 0.8$, and $\delta = 0.3$. We compared its results to those from IEThresh with $\epsilon = 0.8$ and Repeated on the six UCI data sets. For each collection of labelers, we ran 10 separate test/training combinations as described earlier. Thus we ended up with five sets of curves (one per set of 50 labelers), each an average of 10 runs of the algorithms.

Figure 2 shows[5], for data set kr-vs-kp, the cost required to reach a particular classification accuracy for each of the algorithms ("Baseline" is the classification accuracy of training a naïve Bayes classifier on the entire training set with all labels correctly specified). From the figure, we can see that IEAdjCost with $\lambda \leq 0.5$ requires significantly less cost to achieve high classification accuracies than the other algorithms. IEAdjCost with $\lambda = 0.75$ matched the performance of IEThresh, except for data sets splice and nursery (not shown), where it had an advantage. Repeated was the worst performer of all algorithms on all data sets.

The figure shows how much cost is incurred by each algorithm to achieve a certain accuracy. To instead consider how well each method can perform under a fixed budget, we noted the minimum cost across all algorithms required to achieve the baseline accuracy. (For each dataset, this was achieved by IEAdjCost with $\lambda = 0.02$.) Table II reports this number for each dataset, along with each algorithm's accuracy from spending only that budget.

We found that, at the specified budget levels, the performance improvement of IEAdjCost with $\lambda = 0.02$ over all other algorithms (including $\lambda = 0.25$) is statistically significant at the $p < 0.01$ level for kr-vs-kp and mushroom, at the $p < 0.0001$ level for car and nursery, at the $p < 0.03$ level for splice, and at the $p < 0.04$ level for spambase. IEAdjCost with $\lambda = 0.25$ outperformed all algorithms except IEAdjCost with $\lambda = 0.02$ at the $p < 0.0001$ level for splice, and at the $p < 0.1$ level for kr-vs-kp.

In results not shown, we also set $\epsilon = 0.7$ and $\epsilon = 0.9$, and found that each algorithm's performance was very similar. Again, the advantages of IEAdjCost over IEThresh and Repeated still exist. In other results not shown, we set $\delta = 0.2$ and $\delta = 0.4$. We found that increasing $\delta$ to 0.4 improved IEAdjCost's performance for all values of $\lambda$, and decreasing it to 0.2 made it worse. Future work is to further tune $\delta$ to see how much more performance improvement we can achieve, perhaps dynamically choosing its value.

Looking at Figure 2 and Table II, one might assume that smaller values of $\lambda$ will always yield superior results. That is not necessarily the case. Since $\lambda$ determines what fraction of all the labelers are considered at each stage, the following guidelines apply to setting its value.

1) If one believes that there exists a labeler that is cheap and has high accuracy, then the smaller the $\lambda$ value is, the more cost one can save on exploring. In our experiments of this section, it is the case that some highly accurate labelers have low cost. As expected, low values of $\lambda$ yielded high-accuracy classifiers at a lower cost than larger values of $\lambda$.
2) If one believes that the only highly accurate labelers are expensive, then increasing $\lambda$ will allow our algorithm to consider cheaper alternatives, which when taken collectively will perform as well as the expensive labelers but for less cost. Figure 3 shows an experiment in which 4 labelers had accuracy 0.8 and cost 100, and 46 labelers had accuracy 0.7 and cost 1. In this case, IEAdjCost with $\lambda = 0.25$ (instead of $\lambda = 0.02$) performs the best. Note that sometimes although a bigger $\lambda$ can find cheaper labelers, it costs more to find such a combination of cheaper labelers.

In conclusion, all algorithms achieved the baseline accuracy eventually, but some required significantly more cost to do so[6]. When the fifty labeler costs and labeler accuracies are chosen uniformly at random from $\{1, \ldots, 30\}$ and $(0.5, 1]$, respectively, then our algorithms clearly required less cost to achieve high classification accuracies than IEThresh and Repeated. Varying $\epsilon$ did not seem to affect the results very much, though larger values of $\delta$ did improve our algorithms' performance somewhat.

*2) Nonuniform Accuracies:* In the second set of experiments, we again used 50 labelers, but designated a certain subset of them to be "good" labelers and the remainder to be "bad" labelers. The accuracies of the good labelers were randomly chosen from $[0.8, 1]$ while the accuracies of the bad labelers were randomly chosen from $(0.5, 0.7]$. The number of good labelers varied from the set $\{10, 20, 30, 40, 50\}$. As

[5]We got similar results for all five sets of curves from the five sets of labelers, so we just present one such set in the figure. We also saw similar results across all UCI data sets.

[6]The number of instances labeled by each algorithm were roughly the same across all experiments; cost was the only thing that varied.

Table II

CLASSIFICATION ACCURACY AFTER SPENDING A LIMITED BUDGET, USING 50 LABELERS, EACH WITH A COST FROM $\{1, \ldots, 30\}$ AND ACCURACY FROM $(0.5, 1]$. $R = 0.95$, $\delta = 0.3$ AND $\epsilon = 0.8$. **Boldface** INDICATES A STATISTICALLY SIGNIFICANT ADVANTAGE OF $\lambda = 0.02$ OVER ALL OTHER ALGORITHMS. *Italics* INDICATES A STATISTICALLY SIGNIFICANT ADVANTAGE OF $\lambda = 0.25$ OVER ALL ALGORITHMS EXCEPT $\lambda = 0.02$.

| | Budget | Classification Accuracy | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | IEAdjCost | | | | | IEThresh | Repeated |
| | | $\lambda = 0.02$ | $\lambda = 0.25$ | $\lambda = 0.5$ | $\lambda = 0.75$ | $\lambda = 1$ | | |
| kr-vs-kp | 10038 | **0.88** | *0.76* | 0.68 | 0.68 | 0.67 | 0.68 | 0.67 |
| mushroom | 9219 | **0.95** | 0.9 | 0.89 | 0.89 | 0.87 | 0.89 | 0.87 |
| car | 11719 | **0.94** | 0.75 | 0.75 | 0.75 | 0.74 | 0.75 | 0.74 |
| splice | 13418 | **0.92** | *0.85* | 0.69 | 0.69 | 0.67 | 0.69 | 0.67 |
| nursery | 10897 | **0.91** | 0.74 | 0.73 | 0.73 | 0.71 | 0.73 | 0.71 |
| spambase | 9259 | **0.90** | 0.87 | 0.87 | 0.87 | 0.86 | 0.87 | 0.86 |

Table III

CLASSIFICATION ACCURACY AFTER SPENDING A LIMITED BUDGET, USING 50 LABELERS, EACH WITH A COST FROM $\{1, \ldots, 30\}$. EACH GOOD LABELER HAS AN ACCURACY FROM $[0.8, 1]$ AND EACH BAD LABELER HAS AN ACCURACY FROM $(0.5, 0.7]$. $R = 0.95$, $\delta = 0.3$ AND $\epsilon = 0.8$. **Boldface** INDICATES A STATISTICALLY SIGNIFICANT ADVANTAGE OF $\lambda = 0.02$ OVER ALL OTHER ALGORITHMS EXCEPT FOR $\lambda = 0.25$ FOR 40 AND 50 LABELERS. *Italics* INDICATES A STATISTICALLY SIGNIFICANT ADVANTAGE OF $\lambda = 0.25$ OVER ALL ALGORITHMS EXCEPT $\lambda = 0.02$.

| # Good Labelers | Budget | Classification Accuracy | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | IEAdjCost | | | | | IEThresh | Repeated |
| | | $\lambda = 0.02$ | $\lambda = 0.25$ | $\lambda = 0.5$ | $\lambda = 0.75$ | $\lambda = 1$ | | |
| 10 | 9539 | **0.88** | 0.69 | 0.69 | 0.68 | 0.67 | 0.69 | 0.67 |
| 20 | 8999 | **0.88** | 0.68 | 0.68 | 0.67 | 0.67 | 0.68 | 0.67 |
| 30 | 10575 | **0.88** | *0.81* | 0.67 | 0.67 | 0.67 | 0.67 | 0.67 |
| 40 | 9058 | **0.88** | *0.87* | 0.68 | 0.68 | 0.67 | 0.68 | 0.67 |
| 50 | 9699 | **0.88** | *0.88* | 0.67 | 0.67 | 0.67 | 0.67 | 0.67 |

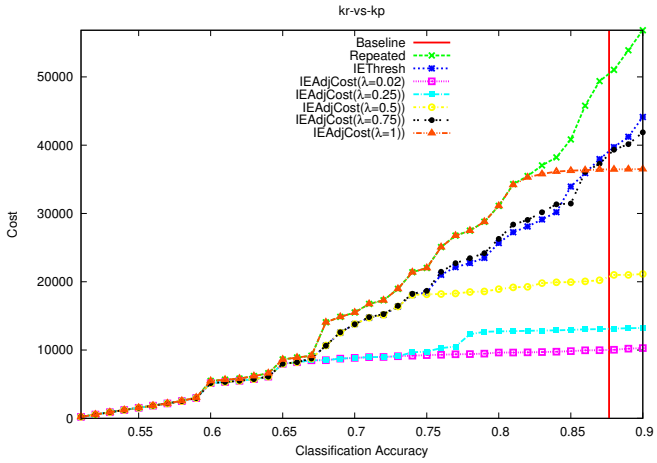


Figure 2. Total cost required for each algorithm to achieve specific classification accuracies on the UCI data set kr-vs-kp. Fifty labelers were available, each with a cost randomly chosen from $\{1, \ldots, 30\}$ and accuracy randomly chosen from $(0.5, 1]$. Values of parameters were $R = 0.95$, $\delta = 0.3$, and $\epsilon = 0.8$.

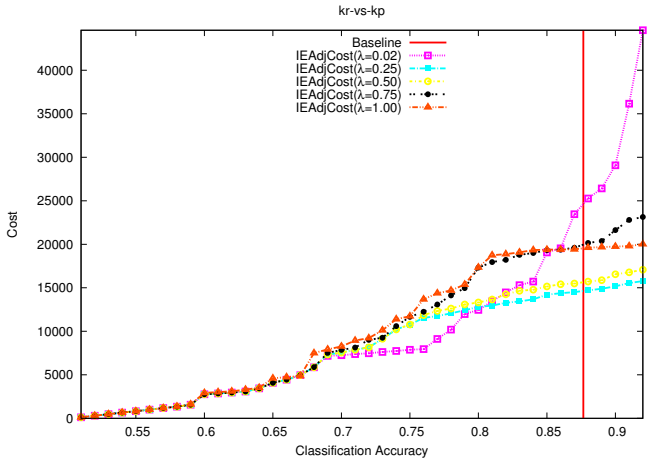


Figure 3. Total cost required for each algorithm to achieve specific classification accuracies on the UCI data set kr-vs-kp. Fifty labelers were available, 4 with accuracy 0.8 and cost 100, 46 with accuracy 0.7 and cost 1. Values of parameters were $R = 0.95$, $\delta = 0.3$, and $\epsilon = 0.8$.

with the previous experiment, the cost of each labeler was randomly chosen from $\{1, \ldots, 30\}$. For IEAdjCost, we used $\epsilon = 0.8$ and $\lambda \in \{0.02, 0.25, 0.5, 0.75, 1\}$.

Table III reports similar information as that in Table II, for varying numbers of good labelers for dataset kr-vs-kp, along with each algorithm's accuracy from spending only that budget (results on the other data sets show similar patterns). We found that, at the specified budget levels, the performance improvement of IEAdjCost with $\lambda = 0.02$ over all other algorithms is statistically significant at the $p < 0.01$ level for all numbers of good labelers (the only exception is that $\lambda = 0.02$ did not have a significant advantage over $\lambda = 0.25$ for 40 and 50 labelers). IEAdjCost with $\lambda = 0.25$ outperformed all algorithms except IEAdjCost with $\lambda = 0.02$ at the $p < 0.01$ level for 30, 40, and 50 good labelers.

### B. Experiments on AMT data sets

We then tested our algorithm on two other data sets: RTE (Recognizing Textual Entailment) and TEMP (Temporal Event Recognition) [9] from Amazon Mechanical Turk (AMT). Each RTE instance is a sentence-sentence pair and the annotators are asked to decide whether the second sentence can be inferred from the first. The original RTE data set has 800 instances and 165 annotators. Each TEMP instance is a short article including two events and the annotators need to judge which of the two events happens first. The original data set has 462 instances and 76 annotators. We used all instances from each data set, but when deciding which labelers to use, we encountered the problem that for each data set, not every annotator labels every instance. Thus we selected as labelers only those labelers who labeled at least 30 instances, leaving 40 labelers for RTE and 31 for TEMP. For each such labeler $o_i$, we used the instances labeled by it to set[7] its accuracy $a_i$. When running the algorithms in our experiments, whenever labeler $o_i$ is asked to label an instance $x$, we first check to see if $o_i$ labeled $x$ in the original data set. In this case, we have $o_i$ return the label that corresponds to it in the original data. If instead $o_i$ did not label $x$ in the original data set, we simulate $o_i$ by having it return the correct label with probability $a_i$. Table IV lists the number of instances and the accuracies of the labelers that remained after this preprocessing step.

Table IV
THE SIZE AND THE LABELER ACCURACIES FOR THE AMT DATA SETS.

| data | size | labeler accuracies |
|---|---|---|
| RTE | 800 | 0.50, 0.51, 0.51, 0.53, 0.56, 0.58, 0.60, 0.65, 0.73, 0.78, 0.80, 0.80, 0.81, 0.82, 0.82, 0.83, 0.83, 0.83, 0.85, 0.85, 0.85, 0.85, 0.85, 0.85, 0.85, 0.88, 0.88, 0.89, 0.90, 0.90, 0.90, 0.91, 0.91, 0.92, 0.92, 0.93, 0.93, 0.93, 0.93, 0.95 |
| TEMP | 462 | 0.44, 0.44, 0.50, 0.54, 0.60, 0.70, 0.77, 0.77, 0.78, 0.80, 0.85, 0.89, 0.90, 0.90, 0.91, 0.92, 0.92, 0.92, 0.92, 0.93, 0.93, 0.93, 0.93, 0.93, 0.93, 0.94, 0.94, 0.95, 0.98, 0.98, 0.98 |

Unlike with the UCI data sets, we did not train classifiers for AMT data because they are sentences instead of attribute vectors. Thus, we adapted the approach used by Donmez et al. [6]: we had each labeler in $O_f$ make its individual prediction based on the procedure outlined above and then output the "classifier's" prediction as a majority vote of these labelers. Hence these experiments evaluate the quality of labelers selected rather than the quality of a classifier induced by training data labeled by the chosen labelers.

As done in Section IV-A, we ran 10 rounds of experiments. In each round we randomly chose 80% of the instances for training data, and used the remaining 20% as testing data. We ran our experiments with two cost models: one with unit cost for each labeler and one with costs randomly selected from $\{1, \ldots, 100\}$. For algorithm IEThresh, we set $\epsilon = 0.8$. For algorithm IEAdjCost, we set $\epsilon = 0.8$, $\delta = 0.3$, and $\lambda \in \{0.025, 0.25, 0.5, 0.75, 1\}$ for RTE and $\lambda \in \{0.03, 0.25, 0.5, 0.75, 1\}$ for TEMP.

[7]This value that we compute as its accuracy is of course just an estimate based on labeled data. But for the purposes of our simulation, we use this value as its "true" accuracy $a_i$.

Table V
EVALUATION OF THE QUALITY OF LABELERS CHOSEN FOR RTE. TRAINING COST IS THE COST OF LABELING ALL TRAINING DATA.

When costs of labelers are 1.

| Method | Training Cost | Testing Accuracy |
|---|---|---|
| Repeated | 25600 | 1.00 |
| IEThresh | 22072 | 1.00 |
| IEAdjCost ($\lambda = 0.025$) | 2342 | 0.98 |
| IEAdjCost ($\lambda = 0.25$) | 2379 | 0.98 |
| IEAdjCost ($\lambda = 0.50$) | 2248 | 0.97 |
| IEAdjCost ($\lambda = 0.75$) | 1568 | 0.93 |
| IEAdjCost ($\lambda = 1.00$) | 2333 | 0.96 |

When costs of labelers are randomly chosen from $\{1, \ldots, 100\}$.

| Method | Training Cost | Testing Accuracy |
|---|---|---|
| Repeated | 1340800 | 1.00 |
| IEThresh | 1172742 | 1.00 |
| IEAdjCost ($\lambda = 0.025$) | 47533 | 0.97 |
| IEAdjCost ($\lambda = 0.25$) | 47533 | 0.97 |
| IEAdjCost ($\lambda = 0.50$) | 45314 | 0.97 |
| IEAdjCost ($\lambda = 0.75$) | 56731 | 0.96 |
| IEAdjCost ($\lambda = 1.00$) | 95620 | 0.94 |

Table VI
EVALUATION OF THE QUALITY OF LABELERS CHOSEN FOR TEMP. TRAINING COST IS THE COST OF LABELING ALL TRAINING DATA.

When costs of labelers are 1.

| Method | Training Cost | Testing Accuracy |
|---|---|---|
| Repeated | 11470 | 1.00 |
| IEThresh | 9674 | 1.00 |
| IEAdjCost ($\lambda = 0.03$) | 1423 | 0.98 |
| IEAdjCost ($\lambda = 0.25$) | 1448 | 0.98 |
| IEAdjCost ($\lambda = 0.50$) | 1424 | 0.98 |
| IEAdjCost ($\lambda = 0.75$) | 1140 | 0.95 |
| IEAdjCost ($\lambda = 1.00$) | 1671 | 0.96 |

When costs of labelers are randomly chosen from $\{1, \ldots, 100\}$.

| Method | Training Cost | Testing Accuracy |
|---|---|---|
| Repeated | 625300 | 1.00 |
| IEThresh | 496520 | 1.00 |
| IEAdjCost ($\lambda = 0.03$) | 29774 | 0.99 |
| IEAdjCost ($\lambda = 0.25$) | 29774 | 0.99 |
| IEAdjCost ($\lambda = 0.50$) | 32959 | 0.99 |
| IEAdjCost ($\lambda = 0.75$) | 42752 | 0.96 |
| IEAdjCost ($\lambda = 1.00$) | 75378 | 0.95 |

Tables V and VI show, for each algorithm, the cost spent on labeling the training data and the accuracy of the selected labelers on the test data for RTE and TEMP. For both data sets, IEAdjCost with $\lambda = 1/|O|$ (0.03 for TEMP and 0.025 for RTE) shows a clear advantage over IEThresh: it saves significant cost on labeling while maintaining high accuracy.

## V. CONCLUSION

We presented a new algorithm IEAdjCost for active learning from multiple labelers with unknown and varied

accuracies and known, varied costs. IEAdjCost has two phases. Phase 1 is similar to IEThresh, which estimates the accuracies of the labelers, except that IEAdjCost reduces costs by reducing the number of labelers used to estimate the true labels and by stopping early when a sufficient number of labelers have sufficiently well-estimated accuracies. In Phase 2, IEAdjCost chooses a subset of the labelers for which a good accuracy estimate exists. When the labelers of this subset are used in a majority voting scheme, their combined accuracy is high and combined cost is low. To achieve this goal, we developed the notions of combined accuracy and adjusted cost to compare labelers with different costs and accuracies. Then our algorithm uses the heuristic LabelersByAdjCost to find the high-accuracy, low-cost subset that is used to label instances from that point forward. We tested our algorithm on six UCI data sets and data from Amazon Mechanical Turk, and showed that our new algorithm performed at least as well, and often better than, algorithm IEThresh from the literature and the naïve approach Repeated. This was especially true when there were many highly accurate labelers available, and/or when high combined accuracy was needed.

In future work, we will look at developing more sophisticated methods to learn when to switch from Phase 1 to Phase 2, perhaps by dynamically tuning $\delta$ and $\lambda$. Another direction is to replace majority voting with weighted majority voting in Phase 2. Other possibilities for future work include replacing Equation (1) with entropy-based uncertainty sampling. We also plan to test IEAdjCost on multi-class data sets, and investigate if our ideas from IEAdjCost could be applied to solve the cost-sensitive feature acquisition problem [18]. Finally, it would be interesting to consider the possibility that labelers' accuracies could be affected by exposure to labeled and unlabeled instances, as what is seen to happen to human labelers [19].

## ACKNOWLEDGMENT

This project is funded in part by National Science Foundation grant 0743783.

## REFERENCES

[1] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *Journal of Machine Learning Research*, vol. 2, pp. 45–66, Nov. 2001.

[2] N. Roy and A. McCallum, "Toward optimal active learning through sampling estimation of error reduction," in *Proceedings of the Eighteenth International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 441–448.

[3] T. Luo, K. Kramer, D. B. Goldgof, S. Samson, A. Remsen, T. Hopkins, and D. Cohn, "Active learning to recognize multiple types of plankton," *Journal of Machine Learning Research*, vol. 6, pp. 589–613, April 2005.

[4] "Amazon Mechanical Turk." [Online]. http://mturk.com

[5] V. S. Sheng, F. Provost, and P. G. Ipeirotis, "Get another label? Improving data quality and data mining using multiple, noisy labelers," in *Proc. of the 14th SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2008, pp. 614–622.

[6] P. Donmez, J. G. Carbonell, and J. Schneider, "Efficiently learning the accuracy of labeling sources for selective sampling," in *Proc. of the 15th SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2009, pp. 259–268.

[7] L. Kaelbling, *Learning in Embedded Systems*. MIT Press, 1993.

[8] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. http://archive.ics.uci.edu/ml

[9] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng, "Cheap and fast—but is it good? Evaluating non-expert annotations for natural language tasks," in *Proc. of the Conf. on Empirical Methods in Natural Language Processing*, 2008, pp. 254–263.

[10] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, p. 138, 1977.

[11] A. P. Dawid and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the EM algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 20–28, 1979.

[12] P. Smyth, U. Fayyad, M. Burl, P. Perona, and P. Baldi, "Inferring ground truth from subjective labeling of Venus images," in *NIPS*, 1995, pp. 1085–1092.

[13] V. Raykar, S. Yu, L. Zhao, A. Jerebko, C. Florin, G. Valadez, L. Bogoni, and L. Moy, "Supervised learning from multiple experts: Whom to trust when everyone lies a bit," in *Proc. of the 26th Int. Conf. on Machine Learning*, 2009, pp. 889–896.

[14] P. Donmez and J. G. Carbonell, "Proactive learning: Cost-sensitive active learning with multiple imperfect oracles," in *Proc. of the 17th ACM Conf. on Information and Knowledge Management*, 2008, pp. 619–628.

[15] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proc. of the 17th Ann. Int. ACM SIGIR Conference on Research and Development in Information Retrieval*, 1994, pp. 3–12.

[16] G. Rätsch, T. Onoda, and K.-R. Müller, "Soft margins for Adaboost," *Machine Learning*, vol. 42, no. 3, pp. 287–320, March 2001.

[17] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA Data Mining Software: An Update; SIGKDD Explorations," 2009, vol. 11, Issue 1.

[18] M. desJardins, J. MacGlashan, and K. Wagstaff, "Confidence-Based Feature Acquisition to Minimize Training and Test Costs," in *Proc. of the SIAM Conference on Data Mining*, 2010, pp. 514–524.

[19] X. Zhu, T. Rogers, R. Qian, and C. Kalish, "Humans perform semi-supervised classification too", in *Twenty-Second AAAI Conference on Artificial Intelligence*, 2007, pp. 864-869.