Efficiently Learning the Accuracy of Labeling Sources for Selective Sampling

Pinar Donmez
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213 USA
pinard@cs.cmu.edu

Jaime G. Carbonell
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213 USA
jgc@cs.cmu.edu

Jeff Schneider
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213 USA
schneide@cs.cmu.edu

ABSTRACT

Many scalable data mining tasks rely on active learning to provide the most useful accurately labeled instances. However, what if there are multiple labeling sources ('oracles' or 'experts') with different but unknown reliabilities? With the recent advent of inexpensive and scalable online annotation tools, such as Amazon's Mechanical Turk, the labeling process has become more vulnerable to noise - and without prior knowledge of the accuracy of each individual labeler. This paper addresses exactly such a challenge: how to jointly learn the accuracy of labeling sources and obtain the most informative labels for the active learning task at hand minimizing total labeling effort. More specifically, we present IEThresh (Interval Estimate Threshold) as a strategy to intelligently select the expert(s) with the highest estimated labeling accuracy. IEThresh estimates a confidence interval for the reliability of each expert and filters out the one(s) whose estimated upper-bound confidence interval is below a threshold - which jointly optimizes expected accuracy (mean) and need to better estimate the expert's accuracy (variance). Our framework is flexible enough to work with a wide range of different noise levels and outperforms baselines such as asking all available experts and random expert selection. In particular, IEThresh achieves a given level of accuracy with less than half the queries issued by all-experts labeling and less than a third the queries required by random expert selection on datasets such as the UCI mushroom one. The results show that our method naturally balances exploration and exploitation as it gains knowledge of which experts to rely upon, and selects them with increasing frequency.

Categories and Subject Descriptors

I.5.2 [**Pattern Recognition**]: Design Methodology—*Classifier design and evaluation*; H.2.8 [**Database Applications**]: Data mining

KDD'09, June 28–July 1, 2009, Paris, France.

General Terms

Algorithms, Design, Experimentation, Performance, Measurement

Keywords

noisy labelers, estimation, active learning, labeler selection

1. INTRODUCTION

In many data mining applications, obtaining labels for the training data is an error-prone process. Brodley [2] notes that class noise can occur for several reasons including subjectivity, data-dependent error, inappropriate feature information used for labeling, etc. Inductive learning algorithms aim at maximizing the classification accuracy based on a set of training instances. The maximum accuracy achieved depends strongly on the quality of the labeling process. In the case of multiple labelers (experts), class noise is typically a common problem since the quality of annotations may not be controlled. For instance, multiple experts might not agree on the medical diagnosis of a clinical case, or on the primary topic of a document, even if we hypothesize the existence of a ground (or consensus) truth. In remote-sensing applications, image analysis is often a manual process with subjective labeling by multiple labelers. In on-line labeling marketplaces such as the Mechanical Turk (http://www.mturk.com), we do not know *a priori* the reliability of individual labelers, though a distribution is expected.

Sheng et al. [15] addresses multiple noisy labelers and proposes repeated labeling (obtaining multiple labels from multiple labelers for some or all data points) to improve label and model quality. Their work mainly focuses on tasks where it is relatively cheap to obtain labels compared to the cost of data gathering and preprocessing. They have shown that repeated labeling can be preferable to single labeling in the presence of label noise, especially when the cost of data preprocessing is non-negligible. Despite their insightful analysis, the practical value of repeated labeling varies greatly with different cost models and with different labeler accuracies. For instance, they make the strong and often unrealistic simplifying assumption that all labelers are identical, with the same probability of making a labeling mistake. As briefly noted by the authors [15], unknown differing qualities require more sophisticated strategies to deal with noisy labelers in general. In real life, we face the problem of inferring individual labeler quality in the absence of the gold standard labels. Estimating each labeler's quality

can prove crucial to improve the overall labeling and help mitigate considerably the effect of labeling noise.

In this paper, we address directly the challenge of active labeling with multiple noisy labelers, each of which has unknown labeling accuracy. The goal is to estimate each labeler's accuracy and use the estimates to select the highest quality labeler(s) for additional label acquisition. Labeler accuracy estimation requires a degree of exploration as well as exploitation in the form of single-labeling with the best labeler(s). Hence, a sophisticated learner should acquire labeler and label knowledge through repeated trials, balancing the exploration vs. exploitation tradeoff, by first favoring the former and moving gradually to increasing exploitation. We report on the first multi-labeler active learner with such properties. We use the words 'oracle', 'expert' and external 'labeler' interchangably.

We adopt the Interval Estimation (IE) learning [5, 10] as a building block for our framework. IE attempts to estimate the confidence interval on the expected response of an action and then selects the action with the highest upper confidence interval. In our problem, taking an action corresponds to selecting an oracle to query for labeling and an instance to label. We use the responses of the oracles themselves to evaluate the performance of each oracle. Therefore, inclusion of inferior oracles can dramatically slow convergence. To overcome this issue, we propose a thesholding mechanism (IEThresh) - to filter out inferior oracles early in the process. This helps to narrow down the set of potentially good oracles and improves the estimation accuracy with many fewer exploratory trials.

In order to apply IEThresh to our problem, we need to define an appropriate reward function. The reward of each labeler is directly related to whether the labeler makes a labeling mistake or not. Unfortunately, an exact calculation of the reward function is impossible since the true label is unknown. A natural way to estimate the true label is to take the majority vote among the predicted labels from multiple labelers. In this paper, we assume an individual labeler accuracy is better than random guess, i.e. > 0.5 in the binary case. Under this assumption, it is unlikely that all labelers make a labeling mistake at the same time; hence, the majority label is a close approximation to the true label. The method is robust to occasional errors by the majority vote method as demonstrated on several benchmark datasets. We report results on six (mostly UCI) datasets to which we add random noise to simulate a set of labelers, and we also report results on two datasets annotated with real labelers [19]. For the simulated-error cases, we varied the labeler accuracies to show that our method IEThresh can detect the most accurate ones even among a uniform or skewed mix of good and bad labelers. We compared our method IEThresh to two baselines: asking all labelers (as in [15]) and selecting a random labeler for each instance. Section 4 details our thorough comparison.

The rest of the paper is organised as follows: The next section summarizes the relevant work in the literature. Section 3 describes the Interval Estimate Threshold method. The experimental evaluation is detailed in Section 4. Finally, we offer our conclusions and potential future directions in Section 5.

2. RELATED WORK

Traditional active learning focuses on selecting data to be labeled to improve the model performance. Thus far, traditional active learning assumed that there is a single oracle (expert) that answers every query with the correct label; hence, the label acquisition is a noise-free process. These assumptions lack realism as also noted in our earlier work [3]. We addressed fallible experts together with reluctant and variable-cost ones in a setting called proactive learning as an alternative to traditional active learning. That work assumes two experts with differing costs: e.g. one is the perfectly reliable expert whereas the other is a noisy expert whose reliability is conditioned on the instances. They further assume that the fallible expert provides a confidence score together with the label. The confidence score is used to assess the quality of the expert. The instance-conditional reliability of the fallible expert is estimated via an exploration phase where the most representative instances are queried and the confidence is propagated through the neighbors. The paper provides a decision-theoretic framework to make the optimal instance-expert selection. This paper addresses limitations in that work, including generalizing from two to multiple experts, eliminating the need that one expert be a perfect oracle, and eliminating the need for explicit and reliable self-reporting of labeling confidence levels.

Furthermore, considering the cost factor in data mining applications has recently become increasingly popular. Utility based data mining [12] introduces a general economic setting to formulate a strategy of maximum expected utility data acquisition. Budgeted learning, active feature acquisition, etc. address the notion of costly data acquisition. In budget-constrained learning, the total cost of data elicitation is bounded, and label queries may have non-uniform cost. The goal of the learner is to produce the most accurate model under the budget constraints. Active feature acquisition (AFA) considers data with missing feature values. AFA tries to optimize the improvement in model accuracy at minimum cost via selective feature acquisition [9, 14] Cost-sensitive learning, on the other hand, generally deals with the cost of misclassification but not the cost of labeling. This line of work is complementary to the methods and results presented in this paper, and interesting future work would entail a combined model.

Repeated labeling on the same data point has been considered by [15, 17, 18] because the labels may not be reliable, without estimating reliability of specific points or regions of the instance space or labelers. The focus of [17, 18] is to learn from probabilistic labels in the absence of ground truth in an image processing application. In their task, the domain experts examine an image and provide subjective class labels. They provide a probabilistic framework to model the subjective labeling process and use EM to estimate the model parameters as maximizers of a likelihood function [18]. Sheng et al. [15] relies on an active learning framework that uses repeated labeling and provides conditions where repeated labeling can be effective for improving data quality. Their results point out that repeated labeling can give additional benefit especially when the labeling quality is low. However, their work assumes the same level of accuracy for each labeler. The method in this paper goes beyond their results by relaxing the assumptions of identical labeling error rates among experts and *a priori* knowledge of said error rates. Moreover, our method is adaptive as it transitions gradually from exploration-heavy to exploitation-heavy phases, as knowledge of individual labeler accuracy accrues.

3. ESTIMATING LABELER ACCURACY

In this section, we describe our multi-expert active sampling method, which we call IEThresh. It builds upon Interval Estimation (IE) learning [5, 10] which is useful for addressing the exploration vs. exploitation tradeoff. IE has been used extensively in reinforcement learning for action selection and in stochastic optimization problems. We first explain IE and then discuss how we extend it to learn the best oracle(s) to query, favoring exploration in the early phases and exploitation (least error-prone oracle selection) with increasing frequency.

3.1 Interval Estimation Learning

The goal of IE is to find the action a^* yielding the highest expected reward with as few samples as possible; i.e. $a^* = \arg\max_a E[r(a) \mid a]$. The true expected reward is unknown and must be estimated from observed samples. Before each selection, IE estimates a standard upper confidence interval for the mean reward of each action using the sample mean and standard deviation of rewards received so far using that action:

$$UI(a) = m(a) + t_{\frac{\alpha}{2}}^{(n-1)} \frac{s(a)}{\sqrt{n}} \quad (1)$$

where $m(a)$ is the sample mean for a, $s(a)$ is the sample standard deviation for a, n is the number of samples observed from a, and $t_{\frac{\alpha}{2}}^{(n-1)}$ is the critical value for the Student's t-distribution with $n-1$ degrees of freedom at the $\alpha/2$ confidence level.

IE then selects the action with the highest upper confidence interval. The reason is that such an action has a high expected reward and/or a large amount of uncertainty in the reward. If an action has large uncertainty, it indicates that the action has not been taken with sufficient frequency to yield reliable estimates. Selecting this action performs exploration which will increase IE's confidence in its estimate and has the potential of identifying a high reward action. Selecting an action with a high expected reward performs exploitation. Initially, the intervals are large due to the uncertainty of the reward estimates and action choices tend to be explorative. Over time, the intervals shrink and the choices become more exploitative. IE automatically trades off these two. α is a parameter that weights exploration more strongly when it is small and exploitation more strongly when it is large. $\alpha = 0.05$ is a common reasonable choice.

3.2 Interval Estimate Threshold (IEThresh)

The IE algorithm described above can be adapted to work with multiple noisy oracles. Taking an action corresponds to selecting an oracle to ask for a label in our active learning framework, assuming we have already selected an instance to label. Our framework is flexible to work with any instance selection strategy and any supervised learning method. For simplicity, we select the instance to label via uncertainty sampling [7] and adopt a logistic regression classifier to obtain posterior class probabilities $P(y \mid x)$. The most uncertain instance is selected for labeling:

$$x^* = \arg\max_x (1 - \max_{y \in \{1,0\}} P(y \mid x)) \quad (2)$$

One also needs to estimate a reward function for each oracle based on the labels received. The reward of each oracle should be related to the true label for the queried instance, which is not known. Hence, we need a mechanism to estimate the true label. We use a majority vote among multiple, possibly noisy labelers to infer the true label – which will be correct often, but not always. We propose the following reward function $\hat{r} : K \rightarrow \{0, 1\}$ as a mapping from the set of labelers K to a binary value. It is 1 if the labeler agrees with the majority label $\bar{y}$, and 0 otherwise.

$$\hat{r}(j) = \begin{cases} 1 & \text{if } y_j = \bar{y} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

This reward estimate requires sampling some or all oracles to take the majority vote. Its accuracy depends on how well the majority vote represents the true label. When the individual labeler quality is high, the majority vote is a close estimate of the true label since it is unlikely that a majority of the oracles make a mistake on the same instance. We propose to adopt a threshold on the upper interval to 1) filter out the less reliable oracles from the majority voting, 2) reduce the labeling cost and 3) compute the reliability estimates more efficiently. Given k oracles, we select each oracle a that has an upper bound $UI(a)$ (Equation 1) larger than some fraction of the maximum bound at time t:

$$S_t = \{a | UI(a) >= \epsilon * \max_a UI(a)\} \quad (4)$$

where S_t is the set of selected oracles to be queried for labeling. $0 < \epsilon < 1$ is a parameter tuned on a separate dataset that is not used in the experiments.[1] We note that this may result in choosing different number of oracles each time, biasing towards the more reliable ones as the reward estimates become more accurate. We smooth the confidence interval estimates by initially giving each oracle a reward of 1 and 0. At the first iteration, they have the same upper bound and all oracles are selected. As the bounds tighten, underperforming oracles are filtered out and the reliable ones are selected for labeling. The upper bound can be high because there is either little information about the oracle (high variance) or the entire interval is high and the oracle is good (high mean). It is possible that a previously filtered-out oracle will be selected again if the upper bounds of the remaining oracles lower sufficiently. We give below an outline of how IEThresh works:

1. Initialize samples for each oracle with rewards 1 and 0
2. Fit a logistic regression classifier to training data T
3. Pick the most uncertain unlabeled instance x^* for labeling (Eqn. 2)
4. Compute the upper confidence interval for each oracle (Eqn. 1)
5. Choose all oracles S_t within ϵ of the maximum upper confidence interval (Eqn. 4)
6. Compute the majority vote $\bar{y}$ of the selected oracles S_t
7. Update training data $T = T \cup \{x^*, \bar{y}\}$
8. Add calculated rewards (Eqn. 3) to the samples for S_t
9. Repeat 2-8

[1]We note that a more sophisticated tuning could further improve the results, but our experimental results indicate that a reasonable threshold works quite effectively.

Our empirical evaluation indicates that IEThresh is very effective in filtering out the less reliable oracles early in the process and continues to sample the more reliable ones. Next, we describe our experimental results in detail.

4. EXPERIMENTAL EVALUATION

4.1 Data and Settings

We conducted a thorough analysis on eight benchmark datasets from [1, 13, 19]. Six of these datasets are classification problems with characteristics given in Table 1. If the dataset was not originally binary, we converted it using random partitioning into two classes as described in [13]. We partition each of these datasets into 70%/30% train/test splits. For each dataset, the initial labeled set includes one true positive and one true negative instance so that each method has the same initial performance before active learning. The rest of the training set is used as the unlabeled pool. We compared *IEThresh* with two baselines: asking all the oracles (repeated labeling) as presented in [15] (we refer it as *Repeated*), and asking a randomly chosen oracle (which is referred as *Random*). Each time an unlabeled instance is selected by the active learner, a label is generated according to the true accuracy q of the selected oracles(s), i.e. the true label $y \in \{1, 0\}$ is assigned with probability q and $1 - y$ is assigned with probability $1 - q$. If more than one oracle is chosen, then the majority vote is assigned as the label for that instance (ties are broken randomly). We set the total number of oracles to $k = 10$. After labeling, the instance is added to the training set and the classifier is re-trained on the enlarged set. The classifier is tested on the separate test set every time a new instance is added, and the classification error is reported. The results are averaged over 100 runs.

The remaining two datasets in our experiments are from the natural language understanding tasks introduced in [19]. This collection was created using Amazon's Mechanical Turk (AMT) for data annotation. AMT is an online tool where remote workers are paid to complete small labeling and annotation tasks. We selected two binary tasks from this collection: the textual entailment recognition (RTE) and temporal event recognition (TEMP) tasks. In the former task, the annotator is presented with two sentences for each question. He needs to decide whether the second sentence can be inferred from the first. The original dataset contains 800 sentence pairs with a total of 165 annotators who contributed to the labeling effort. The latter task involves recognizing the temporal relation in verb-event pairs. The annotator decides whether the event described by the first verb occurs before or after the second. The original dataset contains 462 pairs with a total of 76 annotators. For both datasets, the quality (accuracy) of annotators are measured by comparing their annotations with the gold standard labels. Unfortunately, most of the annotators completed only a handful of tasks. Therefore, we selected a subset of these annotators for each dataset such that each annotator has completed at least 100 tasks. They have differing accuracies ranging from as low as 0.44 to over 0.9. We note that this violates our assumption of better-than-random labelers. This is a real-life dataset that is not generated based on our assumptions; hence, it is useful to test the robustness of our approach to these. Due to the lack of a large amount of data, we selected only the instances for which all annotators provided an answer, to enable our method to select one, several or all the annotators,

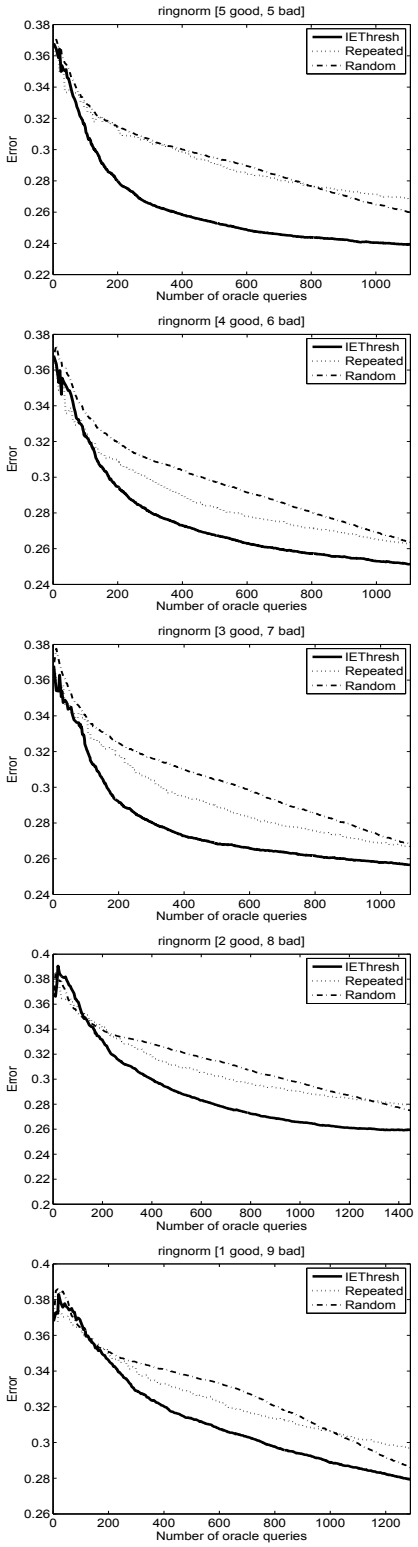

Figure 3: Average classification error vs. total number of oracle queries on *ringnorm* dataset. For the top figure, accuracy $\in [.8, 1]$ for $k_{good} = 5$ labelers and accuracy $\in [.5, .7]$ for the remaining $k_{bad} = 5$ labelers. k_{good} decreases down to 1 and k_{bad} increases up to 9 from top to bottom.

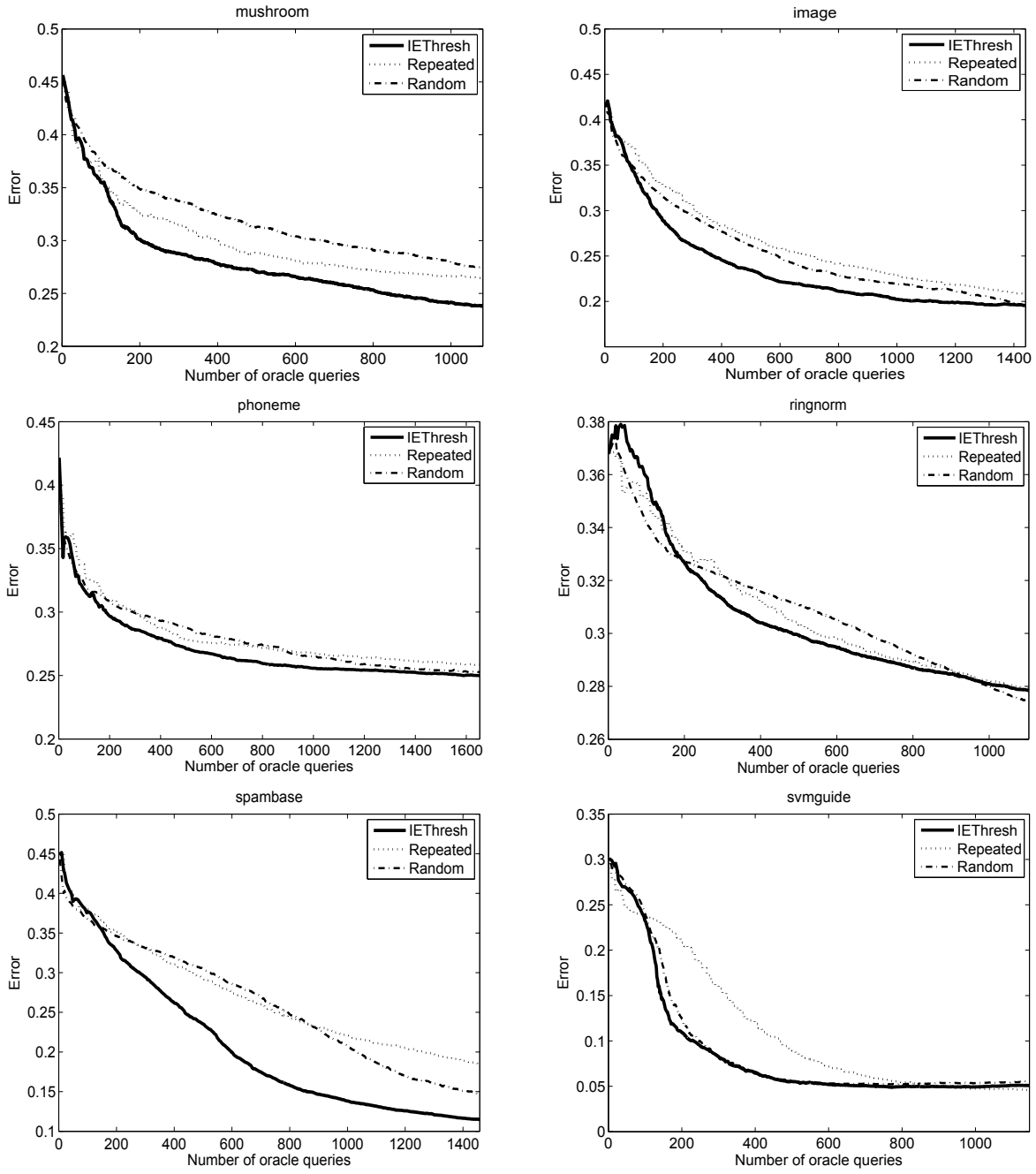

Figure 1: Average classification error vs. total number of oracle queries on six benchmark datasets. Number of oracles is $k = 10$ and the oracle accuracies are selected uniformly at random within the range $[.5, 1]$. The solid curve indicates IEThresh in all graphs. The difference between IEThresh and each baseline is statistically significant ($p < 0.001$) on all datasets based on a two-sided paired t-test at 95% confidence level.

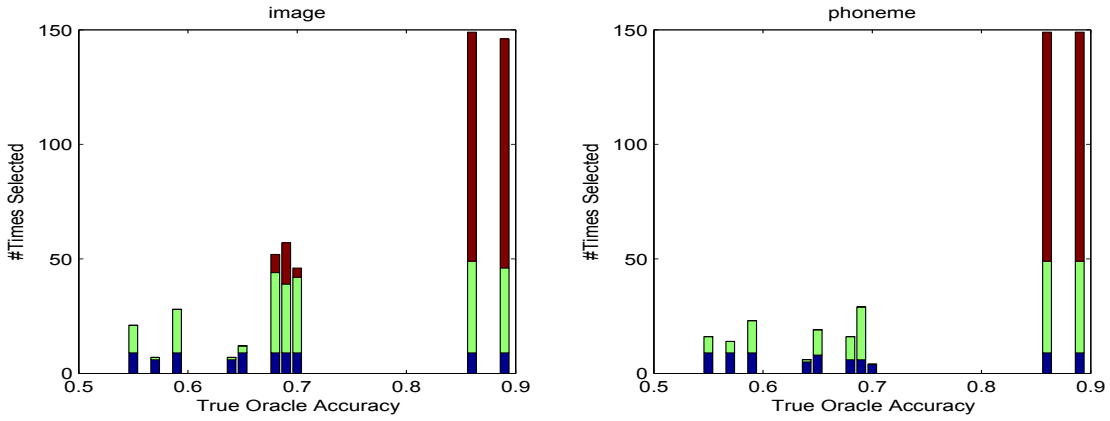

Figure 2: Number of times each oracle is queried vs. the true oracle accuracy. Each oracle corresponds to a single bar. Each bar is multicolored where each color shows the relative contribution. Blue corresponds to the first 10 iterations, green corresponds to an additional 40 iterations and red corresponds to another additional 100 iterations. The bar height shows the total number of times an oracle is queried for labeling by IEThresh during first 150 iterations.

Table 1: Properties of six datasets used in the experiments. All are binary classification tasks with varying sizes.

Dataset	Size	+/- Ratio	Dimensions
image	2310	1.33	18
mushroom	8124	1.07	22
spambase	4601	0.65	57
phoneme	5404	0.41	5
ringnorm	7400	0.98	20
svmguide	3089	1.83	4

and to have consistent baselines. The annotator accuracies and the size of each dataset is reported in Table 2.

We compared our method *IEThresh* against *Repeated* and *Random* baselines on these two datasets. In contrast to the UCI data experiment, there is no training of classifiers for this experiment. Instead, the test set predictions are made directly by AMT labelers. Hence, we randomly selected 50 instances from each dataset to be used by IEThresh to infer estimates for the annotator accuracies. The remaining instances are held out as the test set. The annotator with the best estimated accuracy is evaluated on the test set. The total number of queries are then calculated as a sum of the number of queries issued during inference and the number of queries issued to the chosen annotator during testing. Repeated and Random baselines do not need an inference phase since they do not change their annotator selection mechanism via learning. Hence, they are directly evaluated on the test set. The total number of queries is assigned comparably for IEThresh and Repeated; however, it is equal to the number of test instances for the Random baseline since it queries a single labeler for each instance; thus, there can only be as many queries as the number of test instances.

4.2 Results

Figure 1 compares three methods on six datasets with simulated oracles. The true accuracy of each oracle in Figure 1 is drawn uniformly at random from within the range

Table 2: The size and the annotator accuracies for each AMT dataset.

Data	Size	Annotator Accuracies
TEMP	190	0.44, 0.44, 0.54, 0.92, 0.92, 0.93
RTE	100	0.51, 0.51, 0.58, 0.85, 0.92

Table 3: Performance Comparison on RTE data. The last column indicates the total number of queries issued to labelers by each method. IEThresh performs accurately with comparable labeling effort to Repeated.

Method	Accuracy	# Queries
IEThresh	0.92	252
Repeated	0.6	250
Random	0.64	50

$[.5, 1]$. The figure reports the average classification error with respect to the total number of oracle queries issued by each method. IEThresh is the best performer in all six datasets. In ringnorm and spambase datasets, IEThresh initially performs slightly worse than the other methods, indicating that oracle reliability requires more sampling in these two datasets. But, after the estimates are settled (which happens in $\sim$ 200 queries), it outperforms the others, with especially large margins in spambase dataset. The results reported are statistically significant based on a two-sided paired t-test, where each pair of points on the averaged results is compared.

We also analyzed the effect of filtering less reliable oracles. An ideal filtering mechanism excludes the less accurate oracles early in the process and samples more from the more accurate ones. In Figure 2, we report the number of times each oracle is queried on image and phoneme datasets. The x-axis shows the true accuracy of each oracle. We consider the first 150 iterations of IEThresh and count the number of times each oracle is selected. Each color corresponds to a different time frame; i.e. blue, green and red correspond to

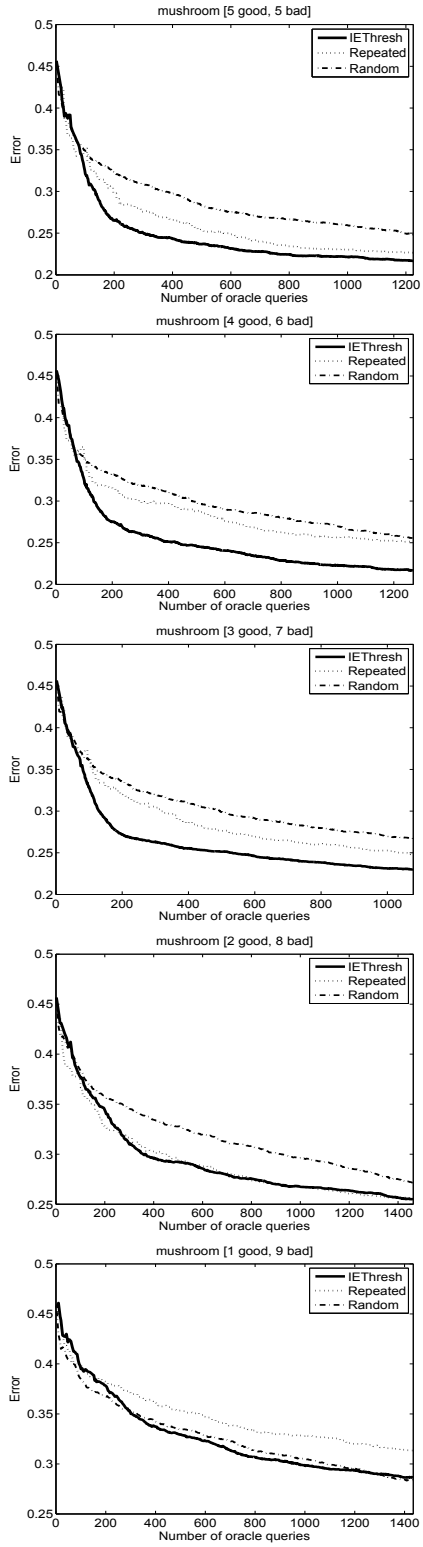

Figure 4: Average classification error vs. total number of oracle queries on UCI *mushroom* dataset. For the top figure, accuracy $\in [.8, 1]$ for $k_{good} = 5$ labelers and accuracy $\in [.5, .7]$ for the remaining $k_{bad} = 5$ labelers. k_{good} decreases down to 1 and k_{bad} increases up to 9 from top to bottom.

Table 4: Performance Comparison on TEMP data. The last column indicates the total number of queries issued to labelers by each method. Repeated needs 840 queries in total to reach 0.95 accuracy to be comparable with IEThresh.

Method	Accuracy	# Queries
IEThresh	0.92	265
Repeated	0.81	280
Random	0.71	140

$0^{th} - 10^{th}$, $10^{th} - 50^{th}$ and $50^{th} - 150^{th}$ iterations, respectively. At first, each oracle is chosen almost equally since the algorithm explores every possibility to improve its estimates. Gradually, we see that less accurate oracles are sampled with decreasing frequency, as reliance shifts to the more accurate ones. The method continues to update its oracle estimates until the estimates converge and become stable.

We further varied distribution of oracle accuracies to challenge IEThresh. Figures 3 and 4 show the resulting performance of each method on ringnorm and mushroom datasets. The top figure on each graph indicates the case with 5 highly fallible oracles with accuracy level within $[.5, .7]$, and 5 reliable ones with accuracies within $[.8, 1]$ range. From the top figure to the bottom, the set of oracles becomes more skewed towards the fallible oracles. The results point out that IEThresh generalizes to work with a wide range of oracle reliability distributions. Even in the challenging case where there are only one or two reliable oracles, the algorithm is able to detect the good ones. Figures 5 and 6 report a similar set of results from a different perspective. The graphs show the total number of queries required to achieve a target classification accuracy. IEThresh requires the least number of queries for a given accuracy level for most cases. Especially when the accuracy targets are high, giving time for IEThresh to stabilize its oracle accuracy estimates, it can improve classification accuracy without the intensive labeling effort required by the baselines. To test the effectiveness of using upper confidence interval in IE learning, we compare with a variant of IEThresh, which we call IEMid, in Figure 6. IEMid considers only the sample mean reward ($m(a)$ in Eqn. 1) of each oracle and selects the oracles whose average reward is larger than a threshold. The results indicate that considering the variance in reward estimates emphasizes better exploration, which is crucial especially when there are only a few good labelers available.

Lastly, we report the results on the RTE and TEMP datasets that have real annotations from multiple less-than-perfect labelers. Table 3 reports the accuracy of each method on the test set for RTE data with the corresponding number of oracle queries issued. The accuracy of IEThresh is the same as the accuracy of the single best labeler in this dataset (See Table 2), indicating that IEThresh managed to detect the best labeler during the inference phase. The Repeated and Random baselines perform poorly in this dataset due to the majority of highly unreliable labelers. Table 4 reports the results on the test set for TEMP data. IEThresh is the best performer in this dataset with a moderate labeling effort. The Repeated labeling baseline needs 840 queries in total to reach 0.95 accuracy. Random baseline stops at 140 queries since this is the size of the test set and it queries a single labeler per instance.

5. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we explored an algorithm for estimating the accuracy of multiple noisy labelers and selecting the best one(s) for active learning. Specifically, we proposed IEThresh as an effective solution that naturally incorporates the exploration vs. exploitation tradeoff. Filtering out the less reliable labelers as early as possible boosts performance. Our experimental evaluation indicates that estimating oracle accuracy, and utilizing these estimates in the active learning process is more effective than the naive counterparts such as asking all labelers or asking a random labeler, which were reported in the recent literature. Even under challenging conditions where the number of reliable labelers is low or some oracles are worse than random (perhaps the AMT labelers misunderstood the instructions), IEThresh is capable of estimating the best labeler(s) through selective sampling and updating oracle accuracy estimates.

There are several directions for expanding the research reported here. One major direction is to track variable oracle performance over time since it could change depending on numerous reasons, e.g. oracle fatigue. For some problems, the labeler quality might go down with extensive labeling due to exhaustion and in some others it might increase with learning. Hence, it is crucial to design methods that goes beyond consistent labeler quality. Another major direction is to condition the probability of making a labeling mistake on the data instance, or at least the region of the instance space which contains the instance. Then, it is crucial to estimate this probability for a representative subset of the input space and generalize to the entire space. Another direction is to relax the assumption that the noise generation is uncorrelated. It is possible that the labelers make correlated errors as noted by [15]. This is a more challenging task since the correlation parameters need to be estimated together with the noise probabilities. Lastly, we assumed that the cost of labeling is the same for each labeler in this paper. However, it is likely that more accurate labelers cost more than the less accurate ones. Furthermore, the cost of each instance might differ according to the difficulty of labeling that instance. In such cases a decision-theoretic utility model would be central. These are interesting and challenging problems that we began investigating under simpler scenarios and plan to investigate in this challenging setting with multiple oracles with *a priori* unknown labeling accuracies.

6. REFERENCES

[1] Blake and C. J. Merz. UCI repository of machine learning databases, 1998.

[2] C. E. Brodley and M. A. Friedl. Identifying and eliminating mislabeled training instances. In *Proceedings of the 13th National Conference on Artificial Intelligence*, pages 799–805, August 1996.

[3] P. Donmez and J. G. Carbonell. Proactive learning: Cost-sensitive active learning with multiple imperfect oracles. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM '08)*, pages 619–628, 2008.

[4] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

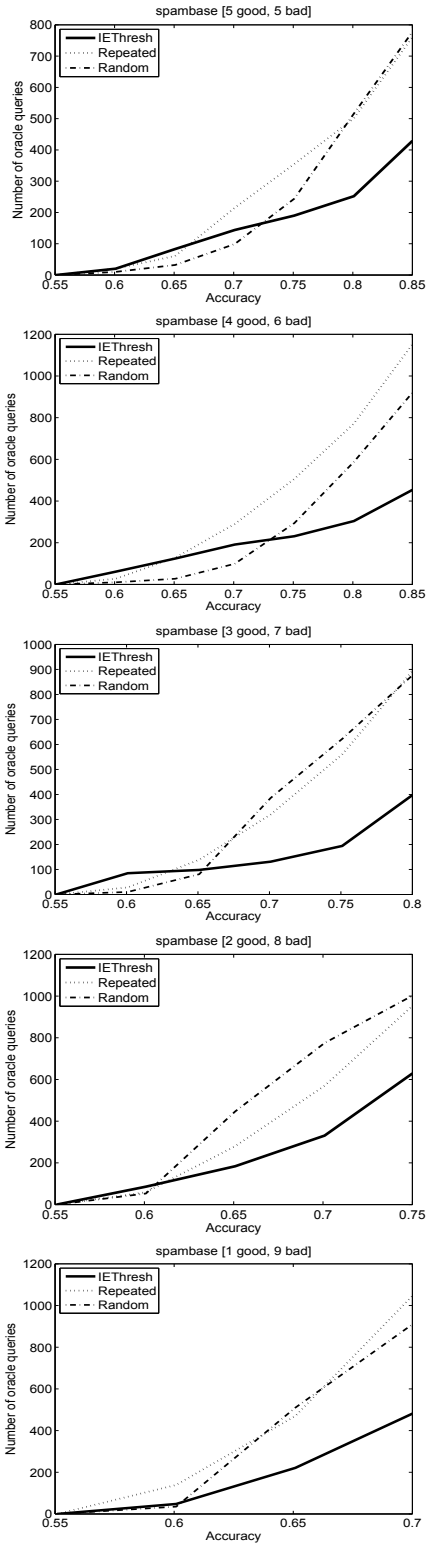

Figure 5: Total number of oracle queries required to reach a target accuracy is plotted on UCI *spambase* dataset. For the top figure, accuracy $\in [.8, 1]$ for $k_{good} = 5$ labelers and accuracy $\in [.5, .7]$ for the remaining $k_{bad} = 5$ labelers. k_{good} decreases down to 1 and k_{bad} increases up to 9 from top to bottom.

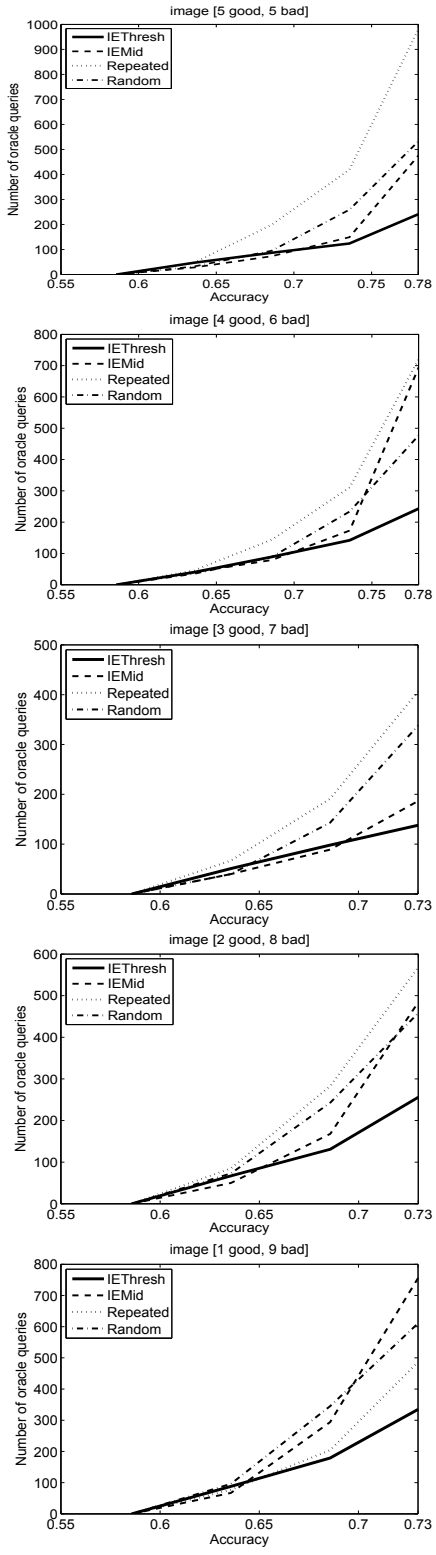

Figure 6: Total number of oracle queries required to reach a target accuracy is plotted on UCI *image* dataset. For the top figure, accuracy $\in [.8, 1]$ for $k_{good} = 5$ labelers and accuracy $\in [.5, .7]$ for the remaining $k_{bad} = 5$ labelers. k_{good} decreases down to 1 and k_{bad} increases up to 9 from top to bottom.

[5] L. P. Kaelbling. *Learning in Embedded Systems*. PhD thesis, Department of Computer Science, Stanford University, 1990.

[6] A. Kappor and R. Greiner. Learning and classifying under hard budgets. In *ECML '05*, pages 170–181, 2005.

[7] D. Lewis and W. Gale. A sequential algorithm for training text classifiers. In *ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12, 1994.

[8] G. Lugosi. Learning with an unreliable teacher. *Pattern Recognition*, 25:79–87, 1992.

[9] P. Melville, M. Saar-Tsechansky, F. Provost, and R. Mooney. An expected utility approach to active feature-value acquisition. In *Proceedings of the 5th International Conference on Data Mining (ICDM '05)*, 2005.

[10] A. Moore and J. Schneider. Memory-based stochastic optimization. In *Neural Information Processing Systems 8*, 1995.

[11] C. T. Morrison and P. R. Cohen. Noisy information value in utility-based decision making. In *Proc. of the First International Workshop on Utility-based Data Mining UBDM '05*, pages 34–38, 2005.

[12] F. Provost. Toward economic machine learning and utility-based data mining. In *Proceedings of the First International Workshop on Utility-based Data Mining*, pages 1–1, 2005.

[13] G. Rätsch, T. Onoda, and K. R. Muller. Soft margins for adaboost. *Machine Learning*, 42(3):287–320, 2001.

[14] M. Saar-Tsechansky, P. Melville, and F. Provost. Active feature-value acquisition. *Management Sciences*, 2008.

[15] V. Sheng, F. Provost, and P. G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)*, pages 614–622, 2008.

[16] B. W. Silverman. Some asymptotic properties of the probabilistic teacher. *IEEE Transactions on Information Theory*, 26:246–249, 1980.

[17] P. Smyth, U. Fayyad, M. Burl, P. Perona, and P. Baldi. Inferring ground truth from subjective labelling of venus images. In *Advances in Neural Information Processing Systems (NIPS '94)*, pages 1085–1092, 1994.

[18] P. Smyth, U. Fayyad, M. Burl, P. Perona, and P. Baldi. Learning with probabilistic supervision. *Computational Learning Theory and Natural Learning Systems*, 3, 1995.

[19] R. Snow, O'Connor, D. Jurafsky, and A. Ng. Cheap and fast—but is it good? evaluating non-expert annotations for natural language tasks. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2008.

[20] Z. Zheng and B. Padmanabhan. Selectively acquiring customer information: A new data acquisition problem and an active learning-based solution. *Management Science*, 52.