Investigating the Effect of Machine-Crowd-Expert Workflow on Oracle Selection for Utility-Maximization-Based Active Learning

Xiaowei Kuang
David R. Cheriton School of Computer Science
University of Waterloo
x4kuang@uwaterloo.ca

Abstract

We proposed a new workflow for oracle querying in active learning under the finite-pool unlabeled data setting. In our workflow, the learning agent can choose to query multiple oracles with different level of expertises and costs. Using a decision-theoretic framework, the agent incorporates the rewards and the costs of querying each oracle into its decision-making process. Empirical evaluations on real-world datasets illustared the decision-making power of the learning agent using our proposed workflow.

Introduction

Active learning (AL) (Lewis, David D and Gale 1994) is a form of machine learning where the learning algorithm starts from a (usually) small labeled dataset and iteratively queries an external oracle for labels on unlabeled data. The goal of an active learning algorithm is to make as few queries as possible to label the data as correctly as possible and at the same time induce a machine learning classifier that achieves reasonable performance as measured by some criteria. Active learning has been successfully applied in large scale annotation tasks where an enormous amount of unlabeled data is to be labeled and paying human expert annotators to label the data manually is not feasible.

Tradtional active learning algorithms focus on the scenario where there is only one oracle which provides labels to the learner and that the oracle makes no errors. However, this situation is not necessary true in modern times where *crowdsourcing*(Howe 2008) becomes an increasingly popular approach for data annotation. The idea of *crowdsourcing* is to hire workers from online marketplaces such as Amazon Mechanical Turk (AMT) and CrowdFlower to label data in a redundant manner. Although crowdsourcing offers a cheap alternative to hiring domain experts for data annotations, the labels collected from crowd workers can be very noisy because crowd workers have a diverse background and have different levels of expertise when it comes to the annotation tasks. To make matters worse, there exists spammers, who provide random answers to tasks, in those marketplaces, which further complicates the process of inferring correct labels from the collected labels.

The existence of multiple (possibly) imperfect oracles in real-world active-learning-based annotation tasks can compromise, if not paid due attention, the data collection pipeline, the very first step in many data-driven applications. For example, supervised machine learning algorithms relies on the correctness of the labels in the training data to induce a good classifier. If the quality of the labels is compromised, the resulting classifiers might learn the incorrect patterns in the training data caused by the noisy labels and give incorrect predictions when deployed in a real-world setting, which can be catastrophic if the classifiers are deployed in mission-critial systems such as medical alert systems and fire alarm systems.

Many methods have been proposed to deal with the above challenge in active learning. A common theme of these approaches is the use of decision making to determine which oracles to query from during the active learning process. In this paper, we build on prior work from (Nguyen, Wallace, and Lease 2015) and introduce a new decision-making workflow in which the active learning agent queries three different oracles sequentially based on some utility measures for a classification task. The three oracles are played by a machine learning classifier, a group of crowd workers hired from Amazon Mechanical Turk and a domain expert, with increasing cost and level of expertises in the active learning tasks. The research question of this paper is: Can an active learning agent labels an unlabeled dataset using as few budgets as measured by cost as possible by querying the low-cost oracles first for exploration and subsequently querying the costly oracles for exploitation?

The remainder of this paper is structured as follows: We discuss relevant work by previous researchers in the next section, followed by a brief description of our proposed method. Next, we present the experiments that we conducted to evaluate our method before moving on to give a discussion about the advantages and disadvantages of our method. Finally, we summarize our work and propose future work in the concluding section.

Related work

Earlier active learning research focuses on solving the problem of "Which unlabeled example to query about?" under the assumption that the oracles prividing the labels are infallible. Among the many methods that were proposed, *Un-*

certainty Sampling (Lewis and Catlett 1994) and *Query by Committee* (Freund et al. 1997) are the most popular approaches. *Uncertainty Sampling* is a strategy where the learning agent selects the example about which it is most uncertain about to be labeled by the oracles. Despite its simplicity, this strategy has been proven to ba an effective strategy in practice (Lewis, David D and Gale 1994). The *Query by Committee* approach forms a committee of learners and selects the example about which the learners disagree to be labeled by the oracles.

Having identified the existence of multiple imperfect oracles in the active learning setting, many efforts have been made to answer the questions of "Which oracle to query?" or "How to infer the correct labels from multiple noisy labels?" in recent research. We identified three different types of approaches to this problem when reviewing the literature: *Learning-based* approach, *Utility-Maximization* approach and *Task-routing* approach.

In the *Learning-based* approach, a probabilistic framework is used to model the quality of the labels and an optimization objective is formulated to find the bext example to query about and the best oracle to quert from. (Raykar et al. 2010) proposed a method for jointly learning a classifier and infering ground-truth labels from multiple noisy labels. Although their work focues on supervised learning setting, their model for inferring ground truth inspires later work in active learning setting such as that of (Fung 2011). (Fung 2011) proposed a method which estimates the accuracy of the annotators using Expectation-Maximization (EM) algorithm and picks the annotator which has the highest confidence on an unlabeled data point.

In the *Task-routing* approach, a work-assignment strategy based on expertise, availability and cost of an oracle is used to determine which oracle to query at each step. (Wallace et al. 2011) augments the binary labels in supervised machine learning by adding an extra label called "uncertain". Using this addition piece of information given by the oracle and modeling an oracle's expertise using his/her salary, they were able to route easy tasks to novice oracles and difficult tasks to more experienced experts.

The *Utility-Maximization approach* was first proposed in (Roy, Nicholas and McCallum 2001) to select the unlabeled example to query, where utility is defined as reduction of expected errors on the pool of unlabeled data. (Kapoor, Horvitz, and Basu 2007) defined utility to be Value of Information (VOI), which represents the risk reduction and cost of querying due to an unlabeled point, and applied the framework in active learning for voice messages classification. (Donmez and Carbonell 2008) factored in the cost of querying oracles in the calculation of utility by defining utilty to be the uncertainty of the learner subtracted by the cost of a query. (Nguyen, Wallace, and Lease 2015) considers a workflow where the active learning agent either queries a group of non-experts for a label on an unlabeled example or asks the expert to correct the labels previously given by the group of non-experts. They calculated the utility values for both actions and chooses the action that maximizes the utility.

Our work extended the workflow introduced in (Nguyen, Wallace, and Lease 2015) by introducing a new oracle, which is played by a machine learning classifier, to the workflow. In our extension, the active learning agent has the options to query the machine learning classifier for labels on a new unlabeled example, ask a group of non-experts to relabel the examples labeled by the machine learning classifier or ask an expert to correct the labels given by the group of non-experts. The goal of our active learning agent is to label a finite pool of unlabeled data as correctly as possible using as few budgets as possible by querying different oracles strategically. We describe our method in the next section.

Method

We first describe the setting of our problem: Given a small set of labeled data points and a pool of N unlabeld data points $X = \{x_1, x_2, \cdots, x_N\}$ coming from input space $\mathcal{X}$, whose labels $\{y_1, y_2, \cdots, y_N\}$ come from label space $\mathcal{Y}$ and are unknown, we are interested in learning a machine learning classifier $f : \mathcal{X} \rightarrow \mathcal{Y}$ to label the pool of unlabeled data as correctly as possible under finite budget constraint B.

Our active learning agent selects and presents a data point to an oracle for labeling at each step of the active learning process until the budget is exhausted. After the chosen data point is labeled by the selected oracle, the active learning agent adds the data point to the dataset and retrains the machine learning classifier using the new dataset. We consider three types of oracles: machine learning classifiers, crowd workers and experts in our method. We describe these oracles below.

Machine Learning Classifiers as Oracle (Machine Oracle)

A machine oracle is played by a probabilistic machine learning classifier $M : \mathcal{X} \rightarrow \mathcal{Y}$, which is trained on a small set of labeled data. Given a query request x from the active learning agent, the machine oracle responds with a label $y = M(x)$.

Once the classifier is trained, it needs no humans supervision and is fully automated. However, because the amount of training data is small, the resultant classifier may not be very accurate. Therefore, the characteristics of a machine oracle are **low cost**, **high availability** and **noisy labels**. For the rest of the paper, we denote the machine oracle as M. We use the probabilistic output from the machine learning classifier: $P_M(x, y)$ as a model for the accuracy of the machine learning classifier, i.e., we approximate the probability that the true label for an item x is y with the probability that the machine learning classifier predicts y given an item x.

Crowd workers as Oracle (Crowd Oracle)

A crowd oracle is played by a group of K crowd workers recruited from crowdsourcing marketplace. Given a query request x from the active learning agent, each crowd worker provides a label $y_t \in \mathcal{Y}$. The crowd labels $\{y_1, y_2, \cdots y_K\}$ are then provided to the active learning agent. Majority voting is used to infer the correct label y from the set of crowd labels $\{y_1, y_2, \cdots y_K\}$.

Although individual crowd worker might give labels only slightly better than chances in certain tasks, the aggregated label by majority voting is usually reasonably good. Although crowd workers can be recruited with an inexpensive price online, the cost of querying a crowd oracle is still higher than that of querying a machine oracle. Therefore, the charactersitics of a crowd oracle are **medium cost**, **medium availability** and **reasonably good labels**. For the rest of the paper, we denote the crowd oracle as C. Because the crowd oracle is fallible, we need a way to esimate its accuracy. We use the Nave Bayes (Bishop 2007) model for the crowd accuracy model.

$$\begin{aligned} &P(Y^\star = y | Y_1 = y_1, \cdots, Y_K = y_K) \\ &= Z * P(Y^\star = y) \prod_{t=1}^{K} P(Y_t = y_t | Y^\star = y) \end{aligned} \tag{1}$$

where Z is the normalization constant and $P(Y_t = y_t | Y^\star = y) = \frac{\sum \mathbb{1}\{Y_t = y_t, Y^\star = y\} + \alpha}{\sum_{y'_t \in \mathcal{Y}} (\mathbb{1}\{Y_t = y'_t, Y^\star = y\} + \alpha)}$ and α is the Laplace smoothing constant. We estimate the probability that the true label is $y^\star$ given the inferred label $\hat{y}$ with $P_C(y|\hat{y}) = \frac{\sum \mathbb{1}\{\hat{Y} = \hat{y}, Y^\star = y\} + \alpha}{\sum_{y' \in \mathcal{Y}} (\mathbb{1}\{\hat{Y} = y', Y^\star = y\} + \alpha)}$

Domain experts as Oracle (Expert Oracle)

An expert model is played by a domain expert in the task being completed. Given a query request x from the active learning agent, the domain expert responds with the label y. In this paper, we assume for simplicity that the expert oracle is infallible, i.e., the label y is also the ground truth for x [1].

Although domain experts are very accuracte in the task, they are expensive to hire and have low avaialbility. Therefore, the characteristics of an expert oracle are **high cost**, **low availability** and **perfect labels**. For the rest of the paper, we denote the expert oracle as E.

Which oracle to query?

In the decision making step of (Nguyen, Wallace, and Lease 2015), the active learning agent first retrains the machine learning classifier on the labeled data collected so far. The agent then considers querying the crowd oracle for a data point in the unlabeled dataset and estimates the expected loss reduction of acquiring a label on both the labeled and unlabeled dataset weighted by the liklihood of the label. Next, the agent considers querying the expert oracle for a data point in the dataset labeled by the crowd oracle and estimates the expected loss reduction similarly. Finally, the agent selects the action probabilistically according to the amount of expected loss reduction due to the action. We extended this workflow by introducing the machine oracle into the querying pipeline and making it the first option for querying, i.e., the agent first considers querying the machine oracle for a data point in the unlabeled dataset, then considers querying the crowd oracle for a data point in the dataset labeled by the machine oracle and finally considers asking the expert oracle to fix the labels from the crowd oracle. Estimates of reduction of loss in each step are recorded similarly for probabilistically sampling of actions. Pseudocode of our extended decision making algorithm is presented in algorithm 1.

[1]Note that we make this assumption for the sake of simplicty. We are not interested in scenario where a particular oracle is perfect but rather we focus on scenario where different oracles have different levels of expertises.

Algorithm 1 Decision-making

Require: D_M: items labeled by the machine oracle
Require: D_C: items labeled by the crowd oracle
Require: D_E: items labeled by the expert oracle
Require: $L(\hat{y}, y)$: the loss of predicting $\hat{y}$ when the true label is y
Require: $P_M(x, y)$: probability that the true label is y as predicted by the machine oracle
Require: $P_C(y_1, \cdots, y_t, y)$: probability that the true label is y given the crowd's labels $(y_1, \cdots, y_t)$
1: Classifier ← TRAIN(D_M, D_C, D_E)
2: P_U ← PREDICT(Classifier, D_U)
3: CurrentLoss ← EstimateLoss(D_U, D_M, D_C, D_E)
4: **for** $x \in D_U$ **do**
5: $L[M, x] \leftarrow 0$
6: **for** $y \in \mathcal{Y}$ **do**
7: ExpLoss ← EstimateLoss($D_U - x, D_M + (x, y), D_C, D_E$)
8: $L[M, x] \leftarrow L[M, x] + P_U(x, y)$ExpLoss
9: **for** $x \in D_M$ **do**
10: $L[C, x] \leftarrow 0$
11: **for** $y \in \mathcal{Y}$ **do**
12: ExpLoss ← EstimateLoss($D_U, D_M - x, D_C + (x, y), D_E$)
13: $L[C, x] \leftarrow L[C, x] + P_M(x, y)$ExpLoss
14: **for** $x, (y_1, \cdots, y_K) \in D_C$ **do**
15: $L[E, x] \leftarrow 0$
16: **for** $y \in \mathcal{Y}$ **do**
17: ExpLoss ← EstimateLoss($D_U, D_M, D_C - x, D_E + (x, y)$)
18: $L[E, x] \leftarrow P_C(y_1, \cdots, y_K, y)$ExpLoss + L[E, x]
19: Score$[M, x] \leftarrow$ (CurrentLoss $- L[M, x]$)/Cost(M)
20: Score$[D, x] \leftarrow$ (CurrentLoss $- L[D, x]$)/Cost(D)
21: Score$[E, x] \leftarrow$ (CurrentLoss $- L[E, x]$)/Cost(E)

We detail the pseudocode for the procedure *EstimateLoss* in algorithm 2. Note that in line 8 the notation $\hat{y}/(y_1, \cdots, y_K)$ means that the labels in the crowd-oracle-labeled dataset can be inferred labels or raw labels from the group of crowd workers in the crowd oracle. Inferred labels may occur due to the fact that in our decision-making procedure, we simulate the new dataset labeled by the crowd oracle by adding hypothetical label. While the real crowd oracle always provide raw labels from the group of crowd workers, the agent can only guess an inferred label when trying to estimate the expected loss reduction. Therefore, the corresonding equation should be used dependening on the types of the labels in the dataset in line 9 and 10.

In our implementation, we used the sampling bias correction technique as mentioned in (Nguyen, Wallace, and Lease 2015) to correct the sampling bias introduced by the active learning agent selecting data points that it is most uncertain abot for labeling. At each active learning step, we restricted the active learning agents to consider only the top 100 uncertain examples for querying the oracles for speedup.

Algorithm 2 Estimate loss

Require: D_M: items labeled by the machine oracle
Require: D_C: items labeled by the crowd oracle
Require: D_E: items labeled by the expert oracle
Require: $L(\hat{y}, y)$: the loss of predicting $\hat{y}$ when the true label is y
Require: $P_M(x, y)$: probability that that the true label is y
Require: $P_C(y_1, \cdots, y_t, y)$: probability that the true label is y given the crowd's labels $(y_1, \cdots, y_t)$

Classifier $\leftarrow$ TRAIN(D_M, D_C, D_E)
Loss $\leftarrow 0$
for $x \in D_U$ **do**
    $\hat{y} \leftarrow$ PREDICT(Classifier, x)
    Loss $\leftarrow$ Loss $+ \sum_{y \in \mathcal{Y}} P_U(x, y) L(\hat{y}, y)$
for $x, \hat{y} \in D_M$ **do**
    Loss $\leftarrow$ Loss $+ \sum_{y \in \mathcal{Y}} P_M(x, \hat{y}) L(\hat{y}, y)$
for $x, \hat{y}/(y_1, \cdots, y_K) \in D_C$ **do**
    InferredLabel $\leftarrow \hat{y}/$MajorityVoting$(y_1, \cdots, y_K)$
    Loss $\leftarrow$ Loss $+ \sum_{y \in \mathcal{Y}} P_C(y|\hat{y}) L(\hat{y}, y) / \sum_{y \in \mathcal{Y}} P_C(y|y_1, \cdots, y_K) L(\hat{y}, y)$

Experiments

Datasets

Following (Nguyen, Wallace, and Lease 2015), we evaluted our proposed method on four systematic review datasets. These datasets are text documents containing academic papers in a particular fields, which were labeled by experts from the field as being relevant to a study or not. In addition, each document in the dataset was also labeled by at least 5 crowd workers recruited from Amazon Mechanical Turk. During labeling, the annotators are presented with the tiltles and abstracts of the documents. The four datasets have the charactersitics that the labels are binary values $\{0, 1\}$ and that the number of negative examples is far greater than the number of positive examples. Detailed statistics of the datasets can be found in table 1

Experimental Setup

Our experimental setup is similar to that of (Nguyen, Wallace, and Lease 2015), we derived TF-IDF features for each document using the title, abstract and keywords of the documents. For the machine learning classifier used by the machine oracle and the active learning agent, we used the *LogisticRegression* algorithm with L1 penalty as implemented in the *Scikit-learn* python library (Pedregosa et al. 2011).

We split every dataset into three non-overlapping subsets according to a ratio of $0.1 : 0.5 : 0.4$. These three subsets are the training set, active learning set and testing set respectively. The training set is used to train the machine learning classifier used by the machine oracle and the crowd accuracy model. The active learning set is the dataset on which our agent will be trained on. The testing set is used to evaluate the generalization performance of the classifier induced by the active learning agent. We report different oracles's precision and recall on the active learning sets in table 2.

For cost of the different oracles, we set the cost for the machine oracle to be 1 and cost of the crowd oracle with 5 crowd workers as 10 (2 for hiring each crowd worker to label one example). According to (Nguyen, Wallace, and Lease 2015), the average amount of money an expert earns in a hour is 100 times as much as that of a crowd worker. Therefore, we set the cost of the expert oracle to be 200.

Initially, the active learning agent is given a randomly selected set of 100 examples labeled by the machine oracle. This step is taken to bootstrap the learning agent. The agent then repeatedly selects examples to be labeled by the oracles and retrains the machine learning classifier until the budget is exhausted. After each active learning step, we measure: (1) the loss of the induced classifier on the active learning set, and (2) the loss of the induced classifier on the testing set. The first measure is taken to evaluate the primary objective of the learning agent: label the unlabeled dataset as correctly as possible. We measure (2) to evaluate whether the agent induces a good classifier. The loss is calculated as a weighted sum of False Positives and False Negatives:

$$\text{Loss} = \text{FP} + \text{R} \times \text{FN}$$

where R is the tradeoff parameter indicating that the cost of missing a relevant document is as much as the cost of including R irrelevant documents. R is usually set to a number greater than 1 because in systematic review missing a relevant document is much more expensive than including an irrelevant document. In addition, large value of R heavily penalizes the learning agent for giving False Negative outputs.

We compare 6 algorithms in our experiments (4 of the algorithms were compared in the experiments conducted by (Nguyen, Wallace, and Lease 2015) and the rest are introduced in our method):

- **US-Machine**: Use uncertainty sampling to select an unlabeled example and query only the machine oracle for answer
- **US-Crowd**: Use uncertainty sampling to select an unlabeled example and query only the crowd oracle for answer
- **US-Expert**: Use uncertainty sampling to select an unlabeled example and query only the expert oracle for answer
- **US-Crowd+Expert**: Use uncertainty sampling to select an unlabeled example. Then query the crowd oracle for the label and immediately query the expert oracle if the crowd oracle does not privide unanimous labels.
- **Decision Theory**: The decision-theoretic approach using Crowd-Expert workflow as proposed in (Nguyen, Wallace, and Lease 2015)

Dataset	Number of documents	Number of relevant documents	relevant documents (%)
Proton Beam	4,749	243	5.1%
Appendicitis	1,664	242	14.5%
DST	8,071	183	2.3%
Omega3	5,774	310	5.3%

Table 1: Statistics of the datasets used in the experiments

- **Decision Theory MCE**: The decision-theoretic approach using Machine-Crowd-Expert workflow as proposed in our paper.

Experiments were run on a Linux server with Intel(R) Core(TM) i7-6770HQ CPU @ 2.60GHz processors and 16GB of memory.

Results

In this subsection, we focused our discussion on the results obtained from running the active learning algorithms on the Proton Beam and Appendicitis datasets as shown in figure 1 and figure 2 because overall findings can also be observed in the resutls obtained from the rest of the two datasets, which are shown in figure 3 and figure 4.

As observed in both figure 1 (a) and figure 2 (a), **US-Machine** and **US-Crowd** reduce the loss on the unlabeled dataset very quickly and immediately entered a plateau at the beginning. This is as expected because querying the machine oracle and the crowd oracle are both relatively cheap options and the learning agent finishes acquiring labels for all the unlabeld examples very quickly. However, due to noisy labels provided by the machine oracle and the crowd oracle, the resulting labeled datasets has non-zero loss. In particular, the loss of **US-Machine** is higher because the recall of the machine oracle is worse than that of the crowd oracle as shown in table 2.

A second observation is that compared with the **US-Expert** algorithm, our proposed method reduced the loss a lot faster as one can observe the large margin between the red curve and the blue curve before the "50,000" cost mark. After the "50,000" cost mark, the two methods (**US-Expert** and **Decision Theory MCE**) are similar in terms of reducing the loss. Intuitively, our proposed method reduces the cost very quickly at the beginning because the agent can query the machine oracle for a large number of examples at a low cost. With this large pool of labeled data, despite the noisy nature, the induced classifier can predict the remaining unlabeld data reasonably well. In addition, since the machine oracle has labeled a lot of examples, a large part of the loss is due to the noisy labels by the machine oracle. In essence, during this stage, the agent mimics the machine oracle's behaviour and performs exploration. After the "50,000" cost mark, the agent has the option of querying the crowd oracle or the expert oracle for fixing the noisy labels due to the machine oracle. In this stage, the agent focuses on exploitation by acquiring good labels from the two oracles. Interestingly, **US-Crowd+Expert** is also a very good strategy when compared with **US-Expert**. However, if the group of crowd workers agreed to an incorrect label unanimously, **US-Crowd+Expert** will not be able to correct the mistakes. This phenomena is observed in figure 2 (a) where the dark blue curve enters a plateau with non-zero loss.

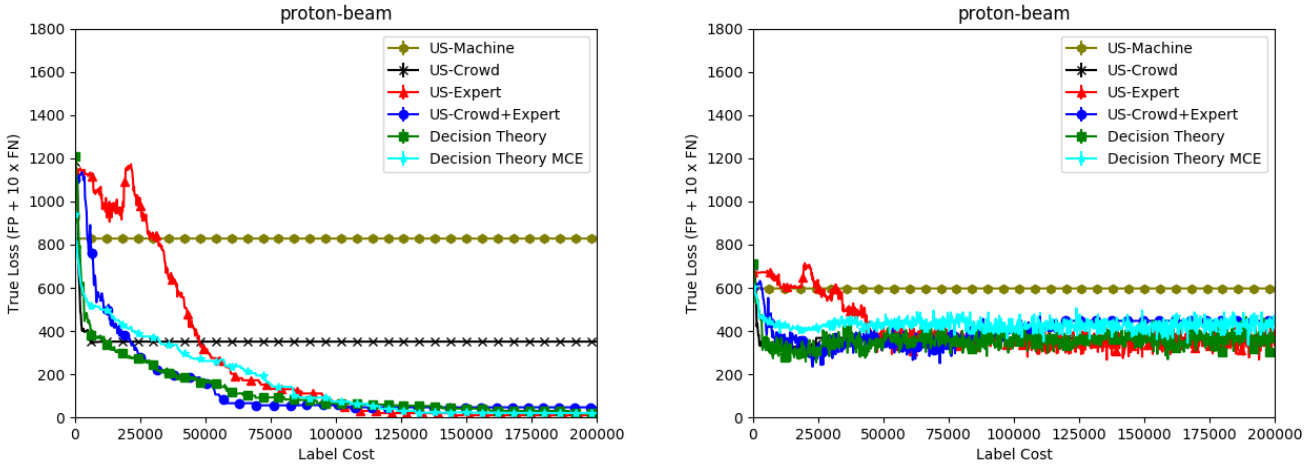
While the loss on the active learning set (part a of the figures) consists of unlabeled loss and the labeled loss due to the labels from the oracles, which obscures performance measures of the induced classifier, figure 1 (b) and figure 2 (b) provide a straightforward measure of the generalization performance of the induced classifier. As shown in the two figures, although the induced classifiers do not generalize well, the observed large margin between the red curve and the light blue curve indicates that the proposed method indeeds learns a better classifier when given the same budget as the **US-Expert** algorithm.

A major motivation for our proposed method comes from the **Decision Theory** method proposed by (Nguyen, Wallace, and Lease 2015). We hypothezied that if a learning agent can make use of the cost difference between the crowd oracle and the expert oracle, then the agent should also be able to exploit the cost differences among the three oracles with different level of expertises. However, although our proposed method outperformed **Decision Theory** on the Appendicitis dataset (figure 2), it is worse than **Decision Theory** on the Proton Beam dataset (figure 1). Similar observations can be seen in figure 3 and figure 4. We suspected there might be several reasons for this: 1) At the time of writing this report, we have only run each experiment once [2], which makes the results susceptible to the influence of randomness. 2) The machine oracle's recall performance is much worse than the crowd oracle's recall performance. On the two datasets (Appendicitis and DST) that our proposed method outperformed the **Decision Theory** method, the differences of recall between the machine oracle and the crowd oracle are around 20%. However, on the other two datasets (Proton Beam and Omega3) where our proposed method performed poorly, the differences are greater than 40%. In such cases, our proposed method suffers because the agent always querys the machine oracle first when trying to label a new unlabeled example and the resulting labels are in some sense a waste of budgets. This suggests that when the responses from one oracle are too noisy compared to other oracles', using that oracle might be a bad idea even though querying costs little.

[2]Running one experiment on one dataset takes apprximately 6 hours on the computer as mentioned in the experimental setup section.

dataset	Machine Oracle		Crowd Oracle		Expert Oracle	
-	Precision (%)	Recall (%)	Precision (%)	Recall (%)	Precision (%)	Recall (%)
Proton Beam (5.11% relevant)	70.00%	34.15%	54.24%	73.17%	100%	100%
Appendicitis (14.45% relevant)	80.43%	30.33%	74.73%	55.74%	100%	100%
Omega 3(5.38% relevant)	72.41%	13.46%	36.54%	70.51%	100%	100%
DST (2.26% relevant)	66.67%	25.81%	52.63%	43.01%	100%	100%

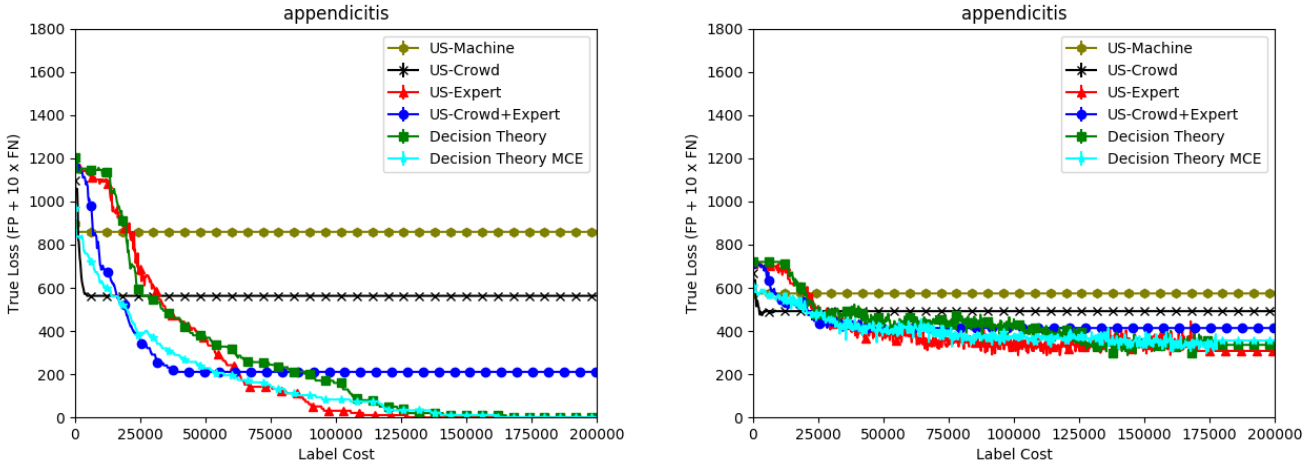
Table 2: Performance of Different Oracles



(a) Loss on active learning set

(b) Loss on testing set

Figure 1: Loss of the induced classifier on the Proton Beam active learning dataset and testing dataset over time. The y-axis represents the weighted loss ($FP + R \times FN$) and the x-axis represents the budget spent by the active learning agent



(a) Loss on active learning set

(b) Loss on testing set

Figure 2: Loss of the induced classifier on the Appendicitis active learning dataset and testing dataset over time. The y-axis represents the weighted loss ($FP + R \times FN$) and the x-axis represents the budget spent by the active learning agent

Discussions

In this section, we discuss advantages and disadvantages of our proposed method.

Advantages

The proposed decision-theoretic method, or broadly, the *Utility-Maximization* approach is very flexible in terms of

incorporating other factors. For example, as in our method, we incorporated consideration of cost of oracles by simply defining utility as a function of the cost. Separation of the decisions about which example and which oracle to query makes incorporation of other modeling techniques in existing methods easy. For example, as mentioned in the related work section, (Fung 2011) proposed a probabilistic model for characterizing crowd workers' abilities. We can easily adopt their method as a more advanced modeling of the crowd workers in our crowd oracle.

Disadvantages

Unlike (Mozafari et al. 2014), which used **Bootstrapping** to turn hard labels from oracles into soft labels, our method requires probabilistic oracles, i.e., oracles which provide probabilistic predictions, which can be a limitation in practice. Our proposed method requires frequent retraining of the classifier at every decision step, which is computationally intensive. This limits the current model to consider only one-step look ahead in the decision-making process, as pointed out in (Nguyen, Wallace, and Lease 2015). In addition, it is difficult to calibrate the loss of the classifier on the datasets with the cost of the different oracles, i.e., designing a utility function requires extensive engineering efforts.

Conclusion

We introduced a new workflow for decision-theoretic active learning under the setting where there exists multiple oracles with different levels of expertises and costs. In this workflow, an active learning agent makes decisions about which oracles to query using maximum expected utility principle. Empirical evaluations on real world datasets showed that our proposed method is competitive to prior work on active learning with multiple oracles and superior to naive algorithms which query only the most accurate oracle.

In future work, we plan to explore combinations of oracles which have different characteristics and hierachical structures, and more flexible querying strategies.

References

Bishop, C. M. 2007. *Pattern Recognition and Machine Learning*, volume 16.

Donmez, P., and Carbonell, J. G. 2008. Proactive Learning : Cost-Sensitive Active Learning with Multiple Imperfect Oracles. *Proceedings of the 17th ACM conference on Information and knowledge management* 619–628.

Freund, Y.; Seung, H. S.; Shamir, E.; and N. Tishby. 1997. Selective Sampling Using the Query by Committee Algorithm. *Machine Learning* 168(1997):133–168.

Fung, G. 2011. Active Learning from Crowds. *Proceedings of the 28th International Conference on Machine Learning* 1161–1168.

Howe, J. 2008. *CROWDSOURCING Why the Power of the Crowd is Driving the Future of Business*.

Kapoor, A.; Horvitz, E.; and Basu, S. 2007. Selective Supervision: Guiding Supervised Learning with Decision-Theoretic Active Learning. In *IJCAI*.

Lewis, D. D., and Catlett, J. 1994. Heterogeneous Uncertainty Sampling for Supervised Learning. In *Machine Learning Proceedings 1994*. Morgan Kaufmann Publishers. 148–156.

Lewis, David D and Gale, W. A. 1994. A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, 3–12.

Mozafari, B.; Sarkar, P.; Franklin, M.; Jordan, M.; and Madden, S. 2014. Scaling up crowd-sourcing to very large datasets. *Proceedings of the VLDB Endowment* 8(2):125–136.

Nguyen, A. T.; Wallace, B. C.; and Lease, M. 2015. Combining Crowd and Expert Labels using Decision Theoretic Active Learning. *Proceedings of the 3rd AAAI Conference on Human Computation (HCOMP)* 120–129.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Raykar, V. C.; Yu, S.; Zhao, L. H.; Hermosillo Valadez, G.; Florin, C.; Bogoni, L.; Moy, L.; and Org, L. M. 2010. Learning From Crowds. *Journal of Machine Learning Research* 11:1297–1322.

Roy, Nicholas and McCallum, A. 2001. Toward optimal active learning through monte carlo estimation of error reduction. In *Proceedings of the Eighteenth International Conference on Machine Learning*, 441–448.

Wallace, B. C.; Small, K.; Brodley, C. E.; and Trikalinos, T. A. 2011. Who Should Label What? Instance Allocation in Multiple Expert Active Learning. In *Proceedings of the 2011 SIAM International Conference on Data Mining*, 176–187.

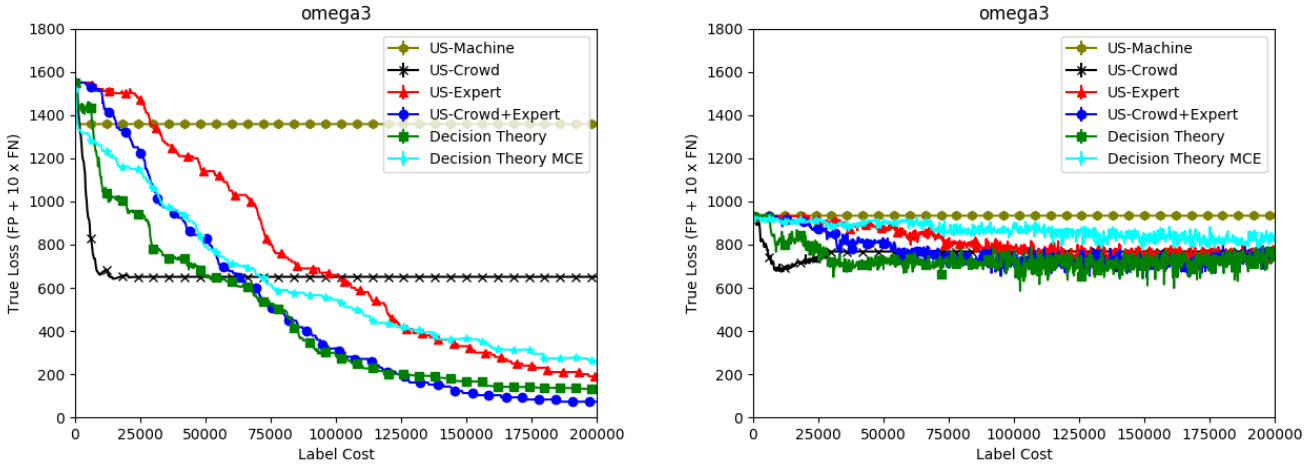


Figure 3: Loss of the induced classifier on the Appendicitis active learning dataset and testing dataset over time. The y-axis represents the weighted loss ($FP + R \times FN$) and the x-axis represents the budget spent by the active learning agent

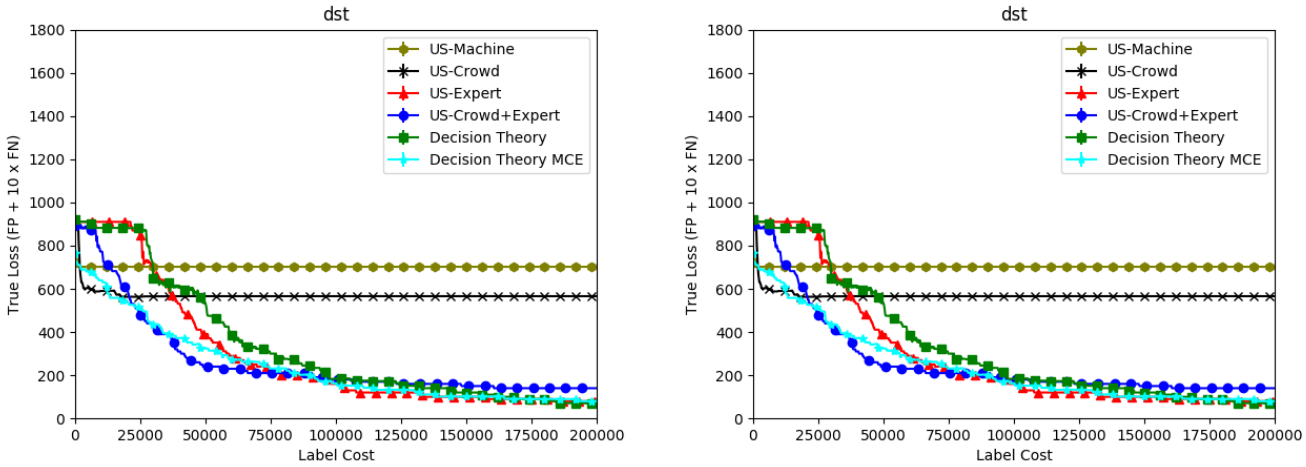


Figure 4: Loss of the induced classifier on the Appendicitis active learning dataset and testing dataset over time. The y-axis represents the weighted loss ($FP + R \times FN$) and the x-axis represents the budget spent by the active learning agent