Modeling Temporal Crowd Work Quality with Limited Supervision

Hyun Joon Jung and Matthew Lease
{hyunJoon | ml}@utexas.edu
School of Information
University of Texas at Austin, USA

Abstract

While recent work has shown that a worker's performance can be more accurately modeled by temporal correlation in task performance, a fundamental challenge remains in the need for expert gold labels to evaluate a worker's performance. To solve this problem, we explore two methods of utilizing limited gold labels, initial training and periodic updating. Furthermore, we present a novel way of learning a prediction model in the absence of gold labels with uncertainty-aware learning and soft-label updating. Our experiment with a real crowdsourcing dataset demonstrates that periodic updating tends to show better performance than initial training when the number of gold labels are very limited (< 25).

Keywords: *crowdsourcing, human computation, prediction, uncertainty-aware learning, time-series modeling*

Introduction

While crowdsourcing offers a cost-efficient and scalable method to collect human labels via the Internet, the quality of work performed by the crowd can greatly vary across individuals, which risks compromising overall data quality. As in traditional employment, a common management strategy is to evaluate the performance of each worker on a regular basis, enabling use of various carrots (e.g., performance-based incentive payments) and sticks (e.g., dismissal of weak performers). In a crowdsourcing context, weighted voting based on individual performance is common. Moreover, if we can accurately predict future job performance based on past evaluations, we can route tasks to those individuals predicted to perform well (Jung 2014), or make proactive interventions before errors occur, helping workers to learn before mistakes are actually made.

Recent work has shown that a worker's performance can be more accurately modeled by abandoning traditional *i.i.d.* assumptions between tasks and instead modeling temporal correlation in task performance (Donmez, Carbonell, and Schneider 2010; Krause and Porzel 2013; Jung, Park, and Lease 2014; Jung and Lease 2015). However, a fundamental challenge remains in the need for expert "gold" labels to evaluate a worker's performance. As Bragg et al. (2014) opined, prior work has often made a strong assumption that all examples have known gold labels readily available to immediately evaluate each worker response as it arrives. Of course, if we already had gold labels in-hand for all examples, there would be no need for collecting additional labels from the crowd.

A common alternative strategy is to ask multiple workers to answer the same question, aggregate responses, and then evaluate each individual's agreement with the aggregate. This poses a fundamental tradeoff in *plurality*: asking more workers to answer the same question increases aggregate accuracy at the cost of increased redundancy. Also, unlike use of expert gold, it cannot safeguard against systematic crowd bias or crowd collusion. Most pertinent in this work, this strategy is difficult to employ in an *online* setting because it is unrealistic to assume that all workers assigned a given example will label it at the same time, or that a worker would happily wait for all others to complete the task before anyone could proceed to the next task (Jung 2014).

We consider how to best estimate a temporal model of worker performance when supervision is more realistically limited. Intuitively, if we have only a smaller sample of gold questions with which to check worker correctness, our estimate of worker accuracy will have larger variance (i.e., increased uncertainty).

Methodology. To solve this problem, we explore how to maximally utilize limited gold labels and how to update a prediction model in the absence of gold labels. Our study begins with an investigation of two method of utilizing limited gold labels. The first method, *initial training* (INIT), uses all of the given gold labels to estimate a worker's label correctness in the initial phase. The second alternative approach, *periodic updating* (PER), uses gold labels to check label correctness periodically. The key insight in periodic updating is that a worker's temporal performance may be non-stationary (ie. exhibiting vayring correctness over time), which may limit the effectiveness of training the model only on the worker's initial temporal patterns.

We also present a novel way of learning a prediction model in the absence of gold labels with *uncertainty-aware learning* (Bootkrajang and Kaban 2013b) and *soft-label updating*. The idea is, for training examples without a known gold label, to generate a pair of positive and negative training examples with *instance* weights based on a probability of the worker producing a correct and incorrect labels. We con-

sider two approaches for estimating uncertainty: one based on model prediction scores and one based on the confidence interval of worker's accuracy.

Finally, our study concludes with an investigation of increasing gold labels vs. use of uncertainty-aware learning with soft labeling. We compare the relative improvement in prediction by adding a single gold label vs. using uncertainty-aware learning.

We evaluate our models on a real crowdsourcing dataset of binary classification. Results demonstrate that *periodic updating* method tends to show better prediction than *initial training* when the number of gold labels are very limited ($<$25). Furthermore, we find that *uncertainty-aware* learning with *soft-label updating* brings substantial improvement to prediction accuracy with limited supervision. Finally, we find that *uncertainty-aware* learning with *soft-label updating* shows substantially higher contribution to the improvement of prediction accuracy than the increase in gold labels does.

We investigate the following research questions:

RQ1: Initial training vs. Periodic updating. *How can we best use limited gold labels for model training? When do different methods perform better?*

RQ2: Uncertainty-aware learning. *To what extent can we effectively update models without supervision? How does uncertainty-aware learning impact prediction accuracy?*

RQ3: Additional gold vs. uncertainty-aware learning. *How does adding a single gold label influence prediction accuracy vs. uncertainty-aware learning with soft labels?*

Problem

We begin with a binary label acquisition problem in crowdsourcing. Suppose that a worker has produced a label set L of n labels ($|L| = n$), and that each label l_i may or may not have a corresponding gold label g_i, which belongs to a gold label set G. Our task is to predict whether or not a worker's next judgment will be correct, as defined by agreement with gold labels. In this work, we assume an objective labeling task for which each example has only a single correct label, indicated by the gold label set.

The correctness of the ith label is denoted as $y_i \in \{0, 1\}$, where 1 and 0 represent correct or not. Label correctness y_i is computed by comparing a worker's label l_i to its corresponding gold label g_i. Thus, the labeling performance of a worker can be represented as a sequence of binary observations, $\mathbf{y} = [y_1 \quad y_2 \quad \dots \quad y_n]$. For example, if a worker produced five labels and erred on the first and third respectively, then her *binary performance sequence* is encoded as $\mathbf{y} = [0 \quad 1 \quad 0 \quad 1 \quad 1]$.

We generate a multi-dimensional feature vector, $x_i = [x_{1i} \quad x_{2i} \quad \dots \quad x_{mi}]$ per time i and use x_i as an input of a prediction function f. We adopt the same features used in (Jung and Lease 2015): observable and latent features about crowd assessors' annotation performance and behavior. However, our feature generation process is different from their study in a sense that feature generation relies upon the availability of gold labels. For instance, when a gold label is provided, we generate the same features as Jung's study. If a gold label is not provided, we include only the subset of features which do not require gold labels to be computed. Specifically, in such cases we omit their accuracy-based features and compute only their behavioral features. Our final goal is to find a prediction function f for each worker and use the function f for predicting each worker's next label correctness.

Prior work has typically assumed the existence of gold labels associated with all of the labels, ($|L| = |G|$) (Donmez, Carbonell, and Schneider 2010; Jung, Park, and Lease 2014; Jung and Lease 2015; Krause and Porzel 2013). However, this assumption does not hold true in practice since one of fundamental reasons for crowdsourcing is collecting labels that we do not have. Furthermore, many studies on online algorithms in quality assurance in crowdsourcing (Tran-Thanh et al. 2014; Welinder and Perona 2010) make a very strong assumption that a worker's label correctness can be checked instantly at each time step (Bragg et al. 2014). We aim to relax this unrealistic assumption by limiting the number of gold labels ($|G| < |L|$) to be used for measuring the label correctness of crowd labels. This is consistent with common practice of injecting occasional questions with known answers into each worker's task queue in order to assess performance. This also resembles a traditional semi-supervised setting in which we seek to learn from unlabeled examples as well as labeled examples, though here we have an additional temporal dimension.

Prior work in item response theory (IRT) (Hambleton, Swaminathan, and Rogers 1991) seeks to assess each individual's temporal learning. However, our approach differs from IRT in that our models seek to capture latent dynamics by taking account of temporal correlation and additional variables. In addition, IRT typically assumes that pairs of questions and answers are provided ahead of a test. These assumptions may not be directly applicable to crowdsourcing settings since if gold labels are in-hand for all examples, collecting additional labels from the crowd serves no useful purpose.

The closest prior work we are aware of on temporal modeling of crowd work with limited supervision, by Krause and Porzel (2013), proposes a method to estimate a worker's response quality by measuring agreement with gold labels as well as using a sliding window over time. However, they assume plurality-based gold estimation, which, as discussed earlier, is difficult to employ in an online setting. Furthermore, this study only leverages given gold labels to estimate worker performance in the beginning while there may exist different ways to use gold labels for worker performance estimation such as periodic checking.

Challenges from limited supervision

Limiting the number of gold labels raises critical questions about how to measure label correctness for model update. Firstly, we face a challenge of measuring label correctness without gold labels. If we assume *offline* analysis (after data collection) and a reasonably high-degree of plurality (the number of worker assigned to the same question), then it is possible to measure a worker' label correctness with *pseudo-gold* labels which can be generated by aggregating multiple labels from workers (Ipeirotis and Provost 2013;

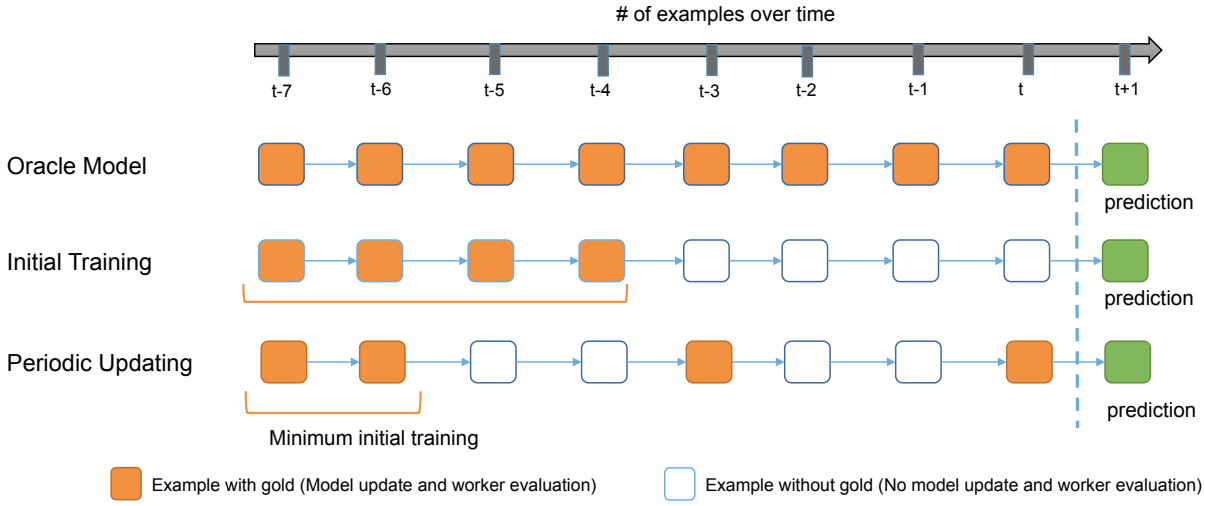

Figure 1: Three sequential learning methods of a prediction model for crowd work quality with limited supervision.

Sheshadri and Lease 2013).

However, when it comes to measuring a worker's label correctness *online* (during data collection), as noted earlier, it is unrealistic to assume that all workers label the same task and they are willing to wait for a next task to be assigned. Moreover, the confidence of pseudo gold labels is sensitive to the number of workers per task. Ideally, we would avoid requiring any plurality and be able to rely upon an individual worker with reliable (predictable) behavior.

Secondly, we should consider how to best use limited gold for training a prediction model. Using all of the given gold labels for *initial training* is simple, but prediction performance may suffer when a worker's temporal performance drifts dynamically over time (non-stationary). A prediction function f trained in this fashion may drift further from the true distribution as the number of labels from this worker $|L|$ increases over time.

Method

We present two methods to address the problems raised in the previous section. Firstly, we explore how to use limited gold for learning a prediction model. Secondly, we introduce a method for learning a prediction model in the absence of gold labels by *soft-label updating*.

Initial Training vs. Periodic Updating

While offline batch learning does not consider the order of training examples, an online learning algorithm is sensitive to order since it processes training examples in a sequential fashion. For this reason, it matters when gold labels are used to check a worker's label correctness with limited gold. In this study, we compare two different methods of using limited gold labels for model training. The first method, *initial training*, uses all of the given number of gold labels G at the start to estimate model parameters. *Initial training* seems appropriate if we assume a sequence of a worker's label correctness follows the property of a stationary process. From a temporal perspective, a sequence of worker's label correctness $\mathbf{y}$ can be described as a stationary process if statistical parameters such as mean, variance, and autocorrelation of $\mathbf{y}$ are all constant over time. However, *initial training*'s prediction performance can be limited if a sequence of a worker's label correctness violates this stationary property.

To relieve this concern, we propose another method, referred to as *periodic updating*, which updates a learning model periodically in order to remain in sync with any temporal drift of a worker's label correctness. Whereas *initial training* uses all k gold labels at the start, *periodic updating* saves limited gold for later checks. In practice, since a learning model requires some amount of training labels in the initial learning phase, *periodic updating* also uses some number of gold labels at the start. However, remaining gold labels are reserved for periodic checking. This method is hypothesized to perform better initial training when worker correctness is not stationary over time.

Figure 1 presents a conceptual example contrasting *initial training* with *periodic updating*. While the former uses 4 gold labels initially, the latter uses two gold labels for periodic updating. As the number of crowd labels increase, the two models are expected to show different performance.

Instance Weighting with Soft Labels

While the two proposed methods in the previous section investigate how to effectively utilize limited gold labels for building a prediction model, due to the absence of gold labels, some labels cannot be checked for their correctness. While it is possible to measure the quality of labels offline via pseudo gold labels (generated by aggregating labels), it is problematic in practice to rely on pseudo-gold labels while data collection is ongoing because workers do not all label the same example at the same time.

Instead, our idea is to estimate and utilize soft labels based on a probability of the worker producing a correct label at time t. For an example with unknown gold, we generate two

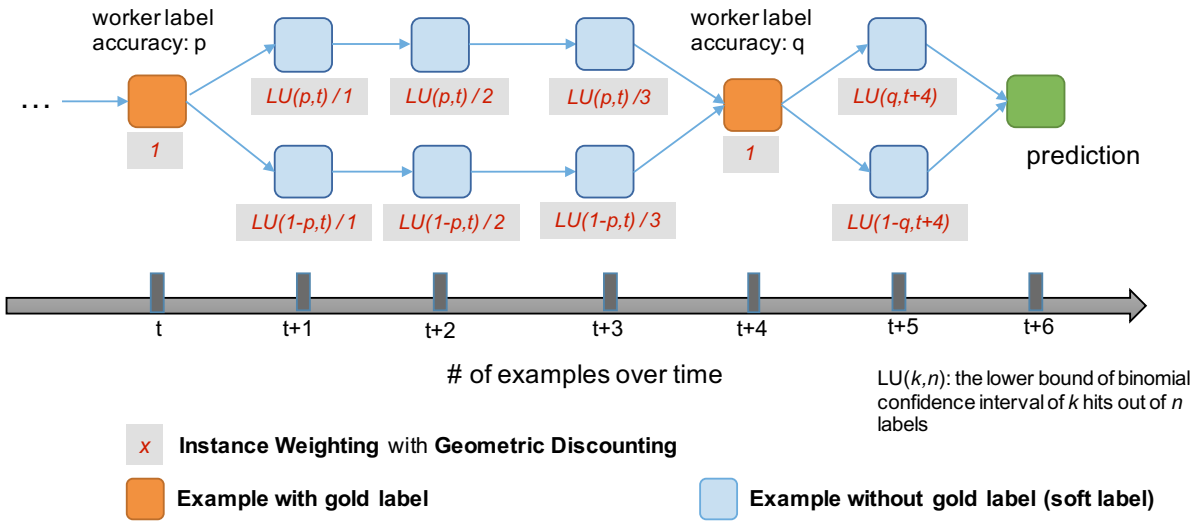

Figure 2: An example of soft labels with lower bound-based approach. Geometric discounting is applied to reduce instance weights with increasing time as a guard against temporal drift.

soft labels: a positive training example and a negative example. We assign instance weights to these training examples: a probability of getting a correct label from this worker at time k, $p(correct)_k$, and 1- $p(correct)_k$.

In order to derive this probability, we first consider a *model score-based* approach, which assigns $p(correct)_k$ and 1- $p(correct)_k$ generated from the model at time t to instance weights at time t+1. Once instance weights are assigned to a pair of two soft labels at time t+1, we update our model and obtain new model scores at time t+1, which are used for instance weights at time t+2.

While the first approach relies upon actual model scores, we consider another way to derive instance weights, a *lower bound-based* approach. In this approach, both the quality of the worker's label accuracy as well as the quantity of labels are considered. Based on the most recent accuracy measured over gold labels, we estimate the probability of a worker producing a correct label. For instance, if the latest worker's accuracy is measured over gold at time t, we use this value for instance weighting.

For lower bound estimation, we adopt the Clopper and Pearson Interval (Clopper and Pearson 1934), the so-called *binomial exact confidence interval*. While the upper bound of it is widely used for exploration and exploitation in Multi-armed Bandit approaches (Auer, Peter and Cesa-Bianchi, Nicolò and Fischer, Paul 2002; Tran-Thanh et al. 2014), we focus on the lower bound of accuracy estimation since our primary goal is to estimate the weight of each training example with minimal uncertainty. The lower bound of the Clopper and Pearson Interval is defined by $B(\frac{\alpha}{2}; x, n-x+1)$ where x is the number of correct labels made by a worker w_j at time i, n is the total number of labels by a worker w_j, and $B(a; b, c)$ is the ath quantile from a beta distribution (b,c).

In addition to lower bound estimation, we also consider *temporal geometric discounting* of the training weight of soft labels. Our intuition is that the confidence of a worker's labeling accuracy at time i diminishes over time given possible non-stationarity in labeling performance. Assuming X_t has a gold label, then for k>0, discount γ_{t+k}= $\frac{1}{k}$ So for k=1,2,3, ... , we have γ_{t+k} = 1, $\frac{1}{2}$, $\frac{1}{3}$, ...

Figure 2 shows an example of training a prediction model with soft labels using *periodic updating*. Whenever a gold label comes, the weight of a training example is reset to 1. However, when no gold is available, we do instance weighting using soft-labels.

Uncertainty-aware Learning

Recent studies in machine learning have investigated how to learn with noisy training examples (Bootkrajang and Kabn 2012; Bootkrajang and Kaban 2013b). We adopt such *uncertainty-aware learning*, which trains a prediction model by including instance weights of each training example.

To select a learning model, we adopt a variant of the *Adaptive Boosting* (Adaboost) model proposed by Bootkrajang and Kaban for several reasons (Freund and Schapire 1997). Firstly, since we need to differentiate the weight of each training example, *weighted Adaboost* exactly fits this need. Secondly, it is a well-known ensemble algorithm that obtains better predictive performance by combining multiple weak learners. Thirdly, a weak learner to be used for this boosting model is logistic regression which is relatively simple to implement and not prone to overfitting. In practice, one of the challenging issues to run learning algorithms online is that it takes too much time to update parameters and predict output values once a new label comes.

In the classical logistic regression model, the log likelihood is defined as:

$$\sum_{n=1}^{N} y_n \log p(y=1|x_n, w) + (1-y_n) \log p(y=0|x_n, w) \quad (1)$$

where w is the coefficient vector. If all of the class labels (y) were presumed to be correct, we would have $p(y = 1|x_n, w) = \sigma(w^T x_n) = \frac{1}{1+e(-w^T x_n)}$ and if this value is greater than 0.5, the predicted value of x_n is class 1. However, when class label noise is present, this approach may not hold true. Thus, *uncertainty-aware learning* introduces a latent variable $\bar{y}$ to consider uncertainty of having an incorrect class label. We model $p(\bar{y} = k|x_n, w)$ as follows:

$$S_n^k \stackrel{\text{def}}{=} p(\bar{y} = k|x_n, w) = \sum_{j=0}^{1} p(\bar{y} = k|y = j)p(y = j|x_n, w) \tag{2}$$

where k $\in 0, 1$. Hence, the log likelihood of *uncertainty-aware* learning is defined as:

$$\sum_{n=1}^{N} \bar{y}_n \log S_n^1 + (1 - \bar{y}_n) \log S_n^0. \tag{3}$$

We omit the details of mathematical proof of this method since it is provided in (Bootkrajang and Kaban 2013a). In prediction, we consider a semi-supervised sequential learning task where we are given N training instances $\{(x_i, y_i), i = 1, ..., N\}$. Here, each $x_i \in \mathbb{R}^M$ is an M-dimensional feature vector adopted from (Jung and Lease 2015), and $y_i \in 0, 1$ is a class label indicating whether an worker's next label is correct (1) or wrong (0). Before fitting a model to our features and target labels, we first normalize feature values using min-max normalization.

Evaluation

Dataset. The NIST TREC 2011 Crowdsourcing Track Task 2 dataset is used[1]. The dataset contains 89,624 *graded relevance judgments (2: highly relevant, 1: relevant, 0: non-relevant)* collected from 762 workers rating the relevance of different Webpages to different search queries (Buckley, Lease, and Smucker 2010). We conflate judgments into a binary scale (relevant / non-relevant). This dataset is processed to extract the original temporal order of the worker's relevance judgments. 3,275 query-document pairs which have expert judgments labeled by NIST assessors are included in the final dataset. In addition, workers making < 20 judgments are excluded; since the goal of our work is to predict worker's next label quality, we intentionally focus on prolific workers expected to continue to work in the future, for whom such predictions will be useful. 49 sequential label sets are obtained, one per worker. The average number of labels (i.e., sequence length) per worker is 134.

Metric. We evaluate the performance of our prediction model with accuracy. Since we build a prediction model per worker, we report mean prediction accuracy across 49 workers. Our extracted dataset is well-balanced in terms of a ratio between relevant vs. non-relevant judgments, and thus use of accuracy is appropriate.

[1] https://sites.google.com/site/treccrowd/

Models. We investigate the prediction accuracy of the proposed methods with a varying number of gold labels. Eight different combinations are considered in this experiment. First, basic *periodic updating* (PER) and *initial training* (INIT) do not update with soft labels. Next, both *periodic updating* with *uncertainty-aware* learning (PER+UNC(MD)) and *initial training* with *uncertainty-aware* learning (INIT+UNC(MD)) update a learning model with soft labels whose instance weights are based on model prediction scores. Finally, *periodic updating* with *uncertainty-aware* learning based on lower bound (PER+UNC(LB)), *periodic updating* with *uncertainty-aware* learning based on lower bound and geometric discounting (PER+UNC(LB+GD)), *initial training* with *uncertainty-aware* learning based on lower bound (INIT+UNC(LB)), and *initial training* with *uncertainty-aware* learning based on lower bound and geometric discounting (INIT+UNC(LB+GD)) use soft labels based on the lower bound of worker accuracy with or without geometric discounting. As a baseline method, we also report an oracle which runs with all known gold labels. We learn a predictive model for each unique label set. To provide minimal training, the first 10 training examples per worker are used for all settings. We evaluate prediction performance by varying the number of gold labels between 11 and 60. While *initial training* uses all k training examples at the start, *periodic updating* uses the remaining k-10 labels for later model updates. Since some worker's number of labels are smaller than 60, we report the number of workers per each number of gold labels in **Table 1**.

For simplicity, *periodic updating* updates the model every $\frac{(n-10)}{(k-10)}$th label with a gold training example (fixed, uniform period update schedule) based on an assumption that the maximum number of labels per worker is n. In practice, when n may not be unknown in advance, it require some exploration for setting the period of model update.

As our base model, we adopt *generalized assessor model* (GAM) (Jung and Lease 2015). While they use *L1-regularized logistic regression*, we instead use a variant of *AdaBoost*, as discussed in the previous section. To learn the AdaBoost model, we use default parameter settings from Scikit-learn (Pedregosa et al. 2011), though setting a learning rate 0.3 after varying parameter values between 0.1 and 1 over the initial training set of each worker.

RQ1: Initial training vs. Periodic updating

What is the best way to utilize limited gold labels for building a more accurate prediction model? We examine the prediction accuracy differences between *initial training* (INIT) and *periodic updating* (PRED). Furthermore, in order to see in what conditions different methods perform better, we investigate the correlation between the extent worker performance is stationary vs. the relative improvement in prediction accuracy by periodic updating.

Table 1 shows mean accuracy of our prediction models across 49 workers. The prediction accuracy is initially measured by each worker and then we compute the overall average score across 49 workers. To examine if the results are

	Number of Gold Examples									
Method	15	20	25	30	35	40	45	50	55	60
Number of Workers (sample size)	49	49	49	49	49	49	49	49	48	47
Model Update Period (labels)	25	12	8	6	5	4	3.5	3	2.7	2.5
0: Oracle	80.7									
1: INIT	56.1	59.6	64.3	66.5	68.4	70.1	73.8	74.5	75.3	75.2
2: PER	58.6*	61.3*	64.4	66.8	68.5	70.5	74.1	74.7	75.7	75.9
3: INIT+UNC(MD)	57.0	60.7	64.2	67.3	68.6	70.2	73.8	74.4	75.4	76.3
4: INIT+UNC(LB)	60.5*	62.8*	65.7	68.3	69.6	71.5	74.3	76.7*	78.4*	79.1*
5: INIT+UNC(LB+GD)	61.3*	63.5*	**67.3***	**69.2***	70.7*	**73.0***	**75.6***	**77.7***	**79.6***	**79.8***
6: PER+UNC(MD)	58.7*	61.4*	64.3	66.9	69.1	70.9	74.1	74.8	76.1	76.5
7: PER+UNC(LB)	61.1*	63.4*	66.4*	68.9*	71.2*	**73.1**	74.6	76.1	77.4	78.4*
8: PER+UNC(LB+GD)	**61.8***	**64.3***	**67.5***	**69.1***	**71.7***	**73.3***	**75.1***	**76.9***	**78.4***	**79.1***

Table 1: Mean prediction accuracy of different prediction models over 49 workers with a varying number of gold training examples. Number of workers per each gold training examples indicates a worker sample size. Model Update Frequency indicates the period of model update which is only applicable to PER. A two-tailed pairwise t-test is conducted to examine whether one model significantly outperforms method 1 (INIT). (*) indicate that one model outperforms method 1 (INIT) with statistical significance ($p<0.05$). Bolded numbers are the best performing methods for each column.

significantly different from each other, we conduct a two-tailed paired t-test. The results show that *periodic updating* outperforms *initial training* when the number of gold labels are very limited (<25). As the number of gold labels increases, the benefit of using periodic updating tends to wane. Considering that the average number of labels per worker is 134, this finding is reasonable. Having 25 gold labels means that *initial training* measures the first 25 labels correctness of a worker and does not update model afterward while *periodic updating* continues the measurement of worker label correctness cyclically with the available number of gold labels. This difference tends to bring a difference of prediction performance when the number of gold labels are significantly limited (<25).

Our idea in periodic updating is that this model would benefit a prediction model for some workers whose distribution of label correctness may not follow a stationary process, which means that mean, variance, and autocorrelation of a worker's label correctness are not constant over time. To investigate this, we compute the autocorrelation of each worker's label correctness by adopting ϕ proposed by Jung, Park, and Lease (2014). Next, the variance of the autocorrelations for each worker is computed since it represents ~~to~~ the extent of being non-stationary over time. Finally, we obtain **Figure 3** which shows a correlation between the variance of autocorrelations vs. relative improvement of prediction accuracy by periodic updating. **Figure 3** shows that periodic updating improves prediction accuracy of workers whose label correctness frequently changes over time (large variance of autocorrelations). This result supports our hypothesis that updating a prediction model periodically with gold examples would improve prediction particularly for workers whose label correctness dynamically drifts over time.

In sum, our first experiment shows that periodic updating improves prediction accuracy when the number of gold labels is very limited (<25). Furthermore, periodic updating is seen to be most valuable when a worker's label quality distribution is not stationary.

RQ2: Uncertainty-aware Learning

The previous experiments demonstrated that periodic updating works more accurately than initial training under limited supervision. Next, how do we update our prediction model when a gold label is not available? And to what extent does this method benefit improving prediction accuracy? We examine the efficacy of *uncertainty-aware learning* with *soft labels* and *geometric discounting* in this experiment.

Table 1 shows that uncertainty-aware learning with soft labels and geometric discounting significantly improves prediction accuracy across a varying number of gold labels. While the effect of this method decreases as the number of gold labels increases, Method 5 (INIT+UNC(LB+GD)) and Method 8 (PRED+UNC(LB+GD)) show substantial improvement of prediction accuracy in comparison to naive initial training (Method 1) and periodic updating (Method 2). In regard to the method of producing soft labels, *lower bound-based* methods outperform *model score-based* methods across initial training and periodic updating. This result suggests that lower bound-based methods tend to provide more accurate soft labels, which leads to greater improvement of prediction accuracy.

Next, we investigate the cause of performance improvement by uncertainty-aware learning with soft-labeling. Our idea in soft labels and instance weighting is that a probability of getting a correct label can be derived from model scores or the lower bound of a worker's accuracy. We expect that if a worker shows low entropy of labeling accuracy (far from accuracy of 0.5), then the benefit of soft labeling increases. To examine this hypothesis, we conduct an additional experiment which focuses on correlation between the prediction accuracy improvement by uncertainty-aware learning vs. workers' label accuracy. To measure the improvement

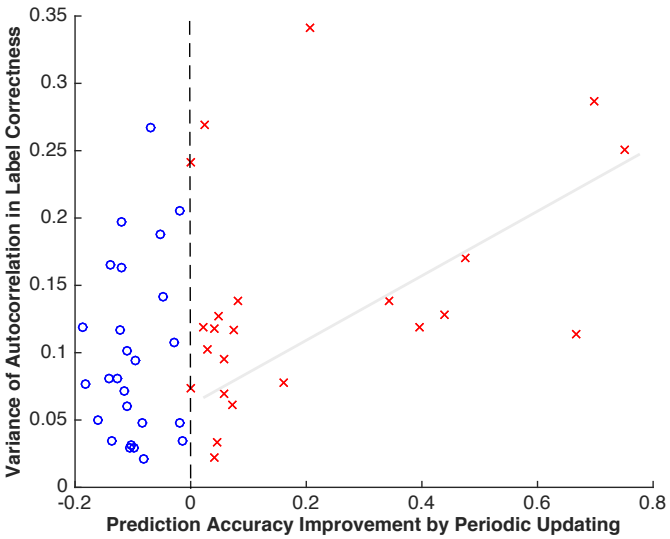

Figure 3: Prediction accuracy improvement by periodic updating vs. variance of crowd label correctness' autocorrelation (number of gold examples = 30). Prediction accuracy improvement by periodic updating is computed by $\frac{PER-INIT}{INIT}$. Both methods work without uncertainty-aware learning. An autocorrelation of label correctness indicates temporal dependency between label correctness, and thus its variance means to the extent temporal dependencies in a sequence of label correctness drifts dynamically over time (non-stationary).

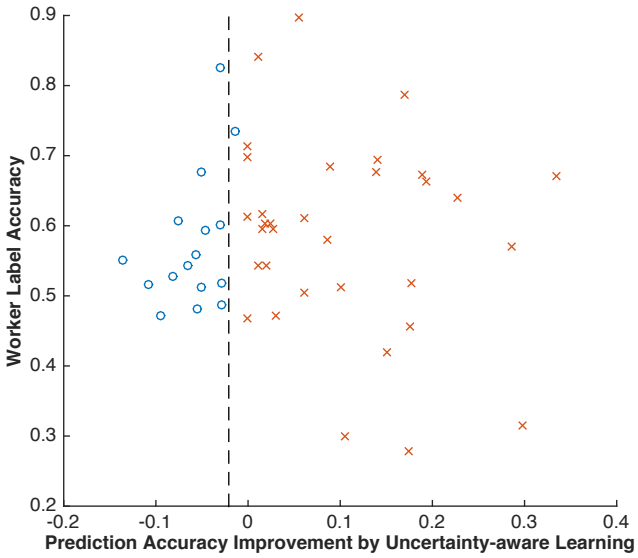

Figure 4: Prediction accuracy improvement by uncertainty-aware learning vs. crowd worker label accuracy (number of gold labels = 30). Prediction accuracy improvement by periodic updating is computed by $\frac{Method8-Method2}{Method2}$. This figure shows that Uncertainty-aware (UNC) learning improves overall prediction accuracy overall. In particular, when label accuracy is reliable (> 0.6 or < 0.4), it is superior to uncertainty free learning.

of prediction accuracy, we compare Method 2, vanilla periodic updating (PER), with Method 8 (PER+UNC(LB+GD)), periodic updating with uncertainty-aware learning based on lower bound and geometric discount. **Figure 4** shows that uncertainty-aware learning achieves better prediction accuracy across 64% (29 out of 45) workers. Note that a prediction model for a worker with accuracy > 0.6 or < 0.4 achieves higher improvement of prediction accuracy. For a worker whose accuracy ranges around 0.5 (higher entropy of labeling accuracy), uncertainty-aware learning based on lower bound and geometric discount tends to show slightly weaker performance improvement.

Finally, we conducted an experiment to investigate how the geometric discount influences prediction accuracy. We hypothesize that if the distribution of a worker's label correctness is non-stationary, the benefit of geometric discount increases. We measure the relative improvement from the geometric discount by comparing Method 7 and Method 8. To measure the degree of being stationary, we measure the variance of a worker's label correctness. **Figure 5** shows when geometric discount brings more benefit in terms of prediction accuracy. This result supports our hypothesis since geometric discount shows bigger improvement for a worker of showing a non-stationary property.

In sum, Experiment 2 demonstrates that uncertainty-aware learning shows substantial improvement of prediction accuracy with soft labels. Furthermore, our additional experiments confirm why and when uncertainty-aware learning with soft labels brings such performance improvement.

RQ3: Additional gold vs. uncertainty-aware learning

Our final experiment seeks an answer to the following question: which method is more efficient to maximize our prediction accuracy: adding one more gold label or applying uncertainty-aware learning? We measure the relative improvement of prediction accuracy at the number of gold labels t and the number of gold labels $t + 1$ between from 10 gold labels and 50 gold labels. At the same time, the benefit of uncertainty-aware learning with soft labels is computed by measuring the difference between method 8 (PER+UNC(LB+GD)) and method 2 (PER) at the number of gold labels t. **Figure 6** shows that while adding one more gold label brings almost 0.5-1% accuracy improvement, uncertainty-aware learning with soft labels improves prediction accuracy up to 5.5%.

Conclusion and Future Work

Limited supervision raises a question of how to learn a model for predicting crowd work quality. For this problem, we explore two methods of using limited gold labels and present a novel way to learn a prediction model with soft labels based on instance weighting. Our experiments with a real crowdsourcing dataset demonstrates that model performance is significantly improved by our proposed methods.

A limitation of this work is its reliance on expert gold for supervision, rather than using peer-agreement between

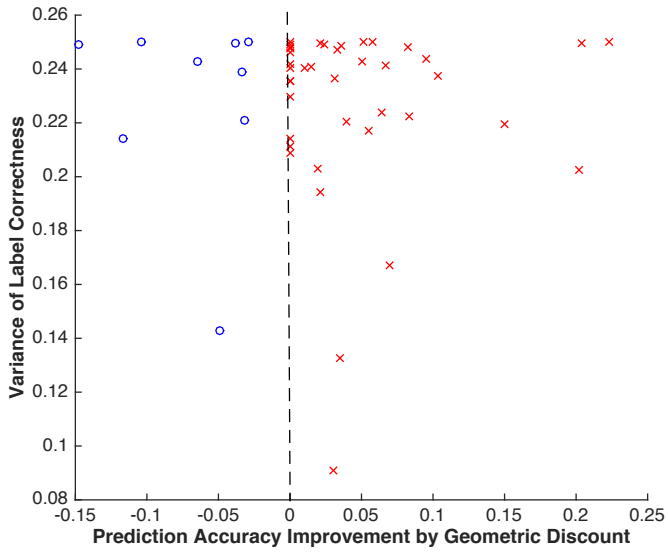

Figure 5: Prediction accuracy improvement by geometric discount vs. variance of label correctness (number of gold labels = 30). Prediction accuracy improvement by geometric discounting is computed by $\frac{Method8-Method7}{Method7}$.

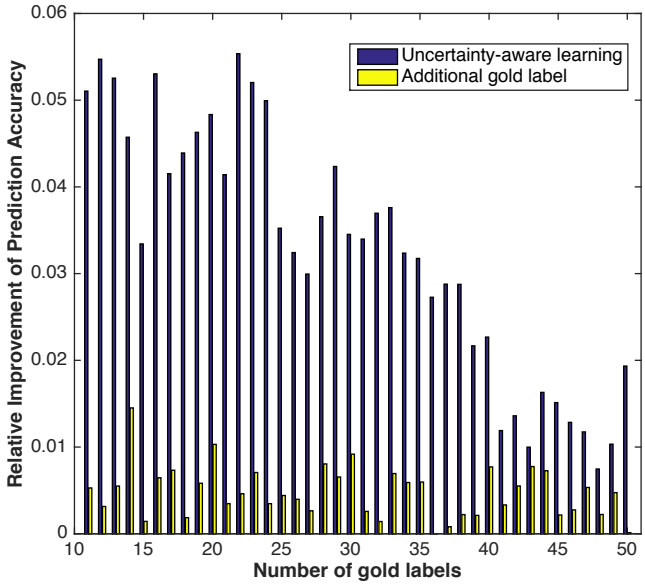

Figure 6: Mean Prediction Accuracy vs. Number of Gold Examples. Y axis indicates the percentage of mean prediction accuracy across 49 workers. Uncertainty-aware methods (INC+UNC, PER +UNC) outperform uncertainty-free methods(INC and PER).

workers to evaluate individual correctness. As discussed earlier, such a strategy is difficult to employ in a live setting because it is unrealistic to assume that all workers label the same example at the same time, or that a worker would happily wait for all others to do so before anyone can proceed to the next task (Jung 2014). To address this challenge, future work will investigate a "lazy update" strategy (Laws, Scheible, and Sch 2011). Instead of updating each worker's model immediately upon label submission, we instead update it later, after all other peer-labels have been received for that example and the consensus label has been established.

As an extension of this study, we aim to predict the average accuracy of the worker in labeling the next k (e.g. 10) tasks. Furthermore, we plan to group a set of workers who show similar labeling performance in order to solve data sparsity (Venanzi et al. 2014).

Finally, there are interesting opportunities to investigate at the intersection of live task-routing with active-learning techniques, specifically in the crowdsourcing context in which we must select both examples to label and workers to do the labeling who offer different cost vs. reliability tradeoffs (Nguyen, Wallace, and Lease 2015).

Acknowledgments. We thank the anonymous reviewers for thoughtful comments and suggestions. We also thank the online crowd contributors who have made this study possible and enabled research on crowdsourcing and human computation to exist and flourish. This study was supported in part by National Science Foundation grant No. 1253413, DARPA Award N66001-12-1-4256, and IMLS grant RE-04-13-0042-13. Any opinions, findings, and conclusions or recommendations expressed by the authors are entirely their own and do not represent those of the sponsoring agencies.

References

Auer, Peter and Cesa-Bianchi, Nicolò and Fischer, Paul. 2002. Finite-time Analysis of the Multiarmed Bandit Problem. *Mach. Learn.* 47(2-3):235–256.

Bootkrajang, J., and Kaban, A. 2013a. Boosting in the presence of label noise. In *Proceedings of the 29rd International Conference on Uncertainty in Artificial Intelligence*, UAI '13.

Bootkrajang, J., and Kaban, A. 2013b. Boosting in the presence of label noise. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence (UAI2013)*.

Bootkrajang, J., and Kabn, A. 2012. Label-noise robust logistic regression and its applications. In Flach, P.; De Bie, T.; and Cristianini, N., eds., *Machine Learning and Knowledge Discovery in Databases*, volume 7523 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg. 143–158.

Bragg, J.; Kolobov, A.; Mausam; and Weld, D. S. 2014. Parallel Task Routing for Crowdsourcing. In *Proceedings of the 2nd AAAI Conference on Human Computation and Crowdsourcing*, HCOMP '14, 11–21.

Buckley, C.; Lease, M.; and Smucker, M. D. 2010. Overview of the TREC 2010 Relevance Feedback Track (Notebook). In *19th Text Retrieval Conference (TREC)*.

Clopper, C. J., and Pearson, E. S. 1934. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika* 26(4):pp. 404–413.

Donmez, P.; Carbonell, J.; and Schneider, J. 2010. A probabilistic framework to learn from multiple annotators with time-varying accuracy. In *Proceedings of the SIAM International Conference on Data Mining*, 826–837.

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

Hambleton, R.; Swaminathan, H.; and Rogers, H. 1991. *Fundamentals of Item Response Theory*. Measurement Methods for the Social Science. SAGE Publications.

Ipeirotis, P., and Provost, F. 2013. Repeated labeling using multiple noisy labelers. *Data Mining and Knowledge Discovery*.

Jung, H. J., and Lease, M. 2015. A Discriminative Approach to Predicting Assessor Accuracy. In *Proceedings of the European Conference on Information Retrieval (ECIR)*.

Jung, H. J.; Park, Y.; and Lease, M. 2014. Predicting Next Label Quality: A Time-Series Model of Crowdwork. In *Proceedings of the 2nd AAAI Conference on Human Computation*, HCOMP '14, 87–95.

Jung, H. J. 2014. Quality Assurance in Crowdsourcing via Matrix Factorization based Task Routing. In *Proceedings of World Wide Web (WWW) Ph.D. Symposium*.

Krause, M., and Porzel, R. 2013. It is About Time: Time Aware Quality Management for Interactive Systems with Humans in the Loop. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, 163–168. New York, NY, USA: ACM.

Laws, F.; Scheible, C.; and Sch, H. 2011. Active Learning with Amazon Mechanical Turk. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 1546–1556.

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

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.

Sheshadri, A., and Lease, M. 2013. SQUARE: A Benchmark for Research on Computing Crowd Consensus. In *Proceedings of the 1st AAAI Conference on Human Computation (HCOMP)*, 156–164.

Tran-Thanh, L.; Stein, S.; Rogers, A.; and Jennings, N. R. 2014. Efficient crowdsourcing of unknown experts using bounded multi-armed bandits. *Artificial Intelligence* 214(0):89 – 111.

Venanzi, M.; Guiver, J.; Kazai, G.; Kohli, P.; and Shokouhi, M. 2014. Community-based Bayesian Aggregation Models for Crowdsourcing. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, 155–164. New York, NY, USA: ACM.

Welinder, P., and Perona, P. 2010. Online crowdsourcing: Rating annotators and obtaining cost-effective labels. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 25–32.