Forecasting Crowd Work Quality via Multi-dimensional Features of Workers

Hyun Joon Jung HYUNJOON@UTEXAS.EDU
School of Information, University of Texas at Austin 1616 Guadalupe St. Austin, TX 78717 USA

Matthew Lease ML@UTEXAS.EDU
School of Information, University of Texas at Austin 1616 Guadalupe St. Austin, TX 78717 USA

Abstract

Modeling changes in individual crowd worker performance over time offers new ways to improve the quality of crowd labels, such as by dynamically routing label annotation tasks to workers more likely to produce reliable labels. Whereas prior crowd annotator models have typically adopted a single generative approach, we formulate a discriminative, flexible feature-based model. This allows us to combine multiple generative models and integrate additional behavioral evidence, enabling better adaptation to temporal variance in worker accuracy. Experiments with a public crowdsourcing data show that our model improves prediction accuracy by 26-36% across workers, enabling 29-47% improved quality of crowd labels to be collected at 17-45% lower cost. Furthermore, we confirm that our proposed model shows significantly accurate prediction than baselines under limited supervision.

1. Introduction

Recent efforts in efficiently collecting labels at scale have focused on how to collect high-quality labels with crowdsourcing (Alonso et al., 2008) (Vuurens & de Vries, 2012) (Lease & Kazai, 2011). Since quality of labels critically influences the performance of learning models (Sheng et al., 2008), a great deal of research has focused quality improvement of crowd labels via various approaches: multiple labeling and aggregation (Venanzi et al., 2014), behavioral effects investigation (Kazai et al., 2012), letting workers select which tasks to work on (Law et al., 2011), and efficient HIT (Human Intelligence Tasks) design (Ipeirotis & Gabrilovich, 2014).

Predicting the quality of labels represents another opportunity to improve quality of crowdsourced labels. For instance, task routing in crowdsourcing (Yuen et al., 2012; Bragg et al., 2014) requires a method to match a worker to a task. One might route a label annotation task to a specific worker based on prediction of that a worker's correctness and expect improved quality of labels as a result.

Prior work in predicting worker' annotation performance has typically relied upon a single generative feature, such as accuracy, with an assumption that crowd labels are independent and identically distributed (i.i.d) over time (Yuen et al., 2012), (Yi et al., 2013). In practice, however, crowd worker behavior can be seen to dynamically vary over time, as shown in **Figure 1**. A worker may become tired or bored, or begin multi-tasking, leading to decreased work quality. Alternatively, work quality may improve as a worker's experience with a given task accumulates (Carterette & Soboroff, 2010). One could imagine many features characterizing such behaviors.

To address this problem, we present a novel discriminative predictive model capturing various behavioral features about a crowd worker, including temporal latent dynamics. This approach allows us to combine multiple generative models and integrate time-series modeling, enabling better adaptation to temporal variance in worker accuracy.

While existing time-series approach seeks to predict a crowd worker's next label more accurately by considering temporal dynamics, it does not consider observable behavioral features about a crowd worker. For this reason, we propose a new *generalized annotator model* (GAM) that utilizes a variety of features to flexibly capture a wider range of worker behaviors to improve prediction performance, as well as the quality of crowdsourced labels. We integrate various features from prior studies which were used only for the estimation of a crowd worker's annotation performance (Ipeirotis & Gabrilovich, 2014) or label simulation (Carterette & Soboroff, 2010). In addition, we devise several new behavioral features indicating a worker's annotation performance over time and integrate them with the existing features selected from prior studies.

Proceedings of the 32^{nd} *International Conference on Machine Learning*, Lille, France, 2015. JMLR: W&CP volume 37.

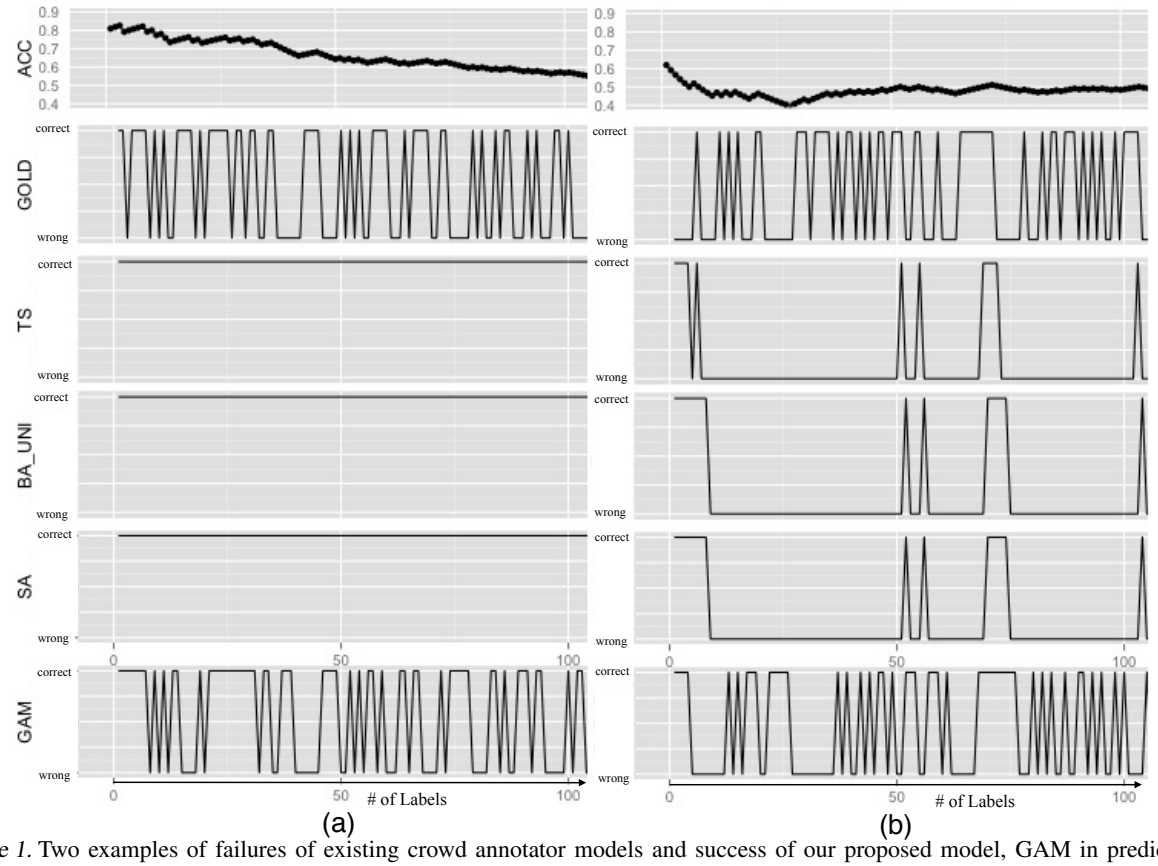

Figure 1. Two examples of failures of existing crowd annotator models and success of our proposed model, GAM in predicting the correctness of workers' next label ((a) high accuracy worker and (b) low accuracy worker). While the agreement of a crowd worker's labels with that of ground truth (GOLD) oscillates over time, the existing worker models (Time-series (TS) (Jung et al., 2014)), Sample Running Accuracy (SA), Bayesian uniform beta prior (BA-UNI (Ipeirotis & Gabrilovich, 2014)) do not follow the temporal variation of the workers' agreement with the gold labels. On the contrary, GAM is sensitive to such dynamics of labels over time for higher quality prediction.

We investigate this predictive model with a public crowdsourcing dataset. Firstly, we evaluate prediction quality, both in terms of hard prediction (binary correct or not) and soft prediction (probability of making a correct label). In particular, we study the effect of a *decision reject option*, which improves prediction accuracy by sacrificing prediction coverage, providing a tuning parameter for aggressive vs. conservative prediction given model confidence. Secondly, we evaluate the effectiveness of our predictive model for crowdsourced label quality improvement under a realistic scenario assuming task routing and label aggregation. Our empirical evaluation demonstrates that our model improves prediction accuracy by 26-36% across 54 workers. In addition, our experiments show that the quality of crowd labels by our prediction model-based task routing improves its accuracy by 29-47% with lower cost (17-45%). Finally, we evaluate the performance of our prediction model under a realisitic condition, which the number of gold labels is limited. Our experiment shows that our model still performs significantly better than the other baselines although its prediction accuracy is impacted by limited supervision. Our research questions are:

RQ1: Prediction Performance Improvement *Can we effectively predict a crowd worker's future work quality? How does decision rejection trade-off coverage vs. accuracy of prediction in comparison to other baselines?*

RQ2: Impact on Label Quality and Cost. *Can work quality prediction be utilized to improve the quality of crowd labels and/or decrease cost of collecting labels?*

RQ3: Impact of Limited Supervision on Prediction Model *How is our prediction model impacted by using only a limited number of training labels?*

2. Problem

In crowdsourcing and human computation, significant research has focused on modeling crowd workers' behavior or performance (Raykar & Yu, 2012) (Rzeszotarski & Kittur, 2011). However, most studies have assumed that each annotation is independent and identically distributed (i.i.d)

over time. In practice, crowd worker behavior can exhibit temporal dynamics, as shown in Figure 1.

SFilter (Donmez et al., 2010), the closest work to our time-series approach, is a Bayesian time-series model that captures crowd workers' dynamically varying performance. However, the authors do not learn the parameters for the latent variable dynamics, but assume an uniform offset and temporal correlation for the underlying dynamics, with workers assumed to be weak learners following simple latent dynamics $x_t = x_{t-1} + \epsilon_t$. Based on the fixed parameters, the latent variable is estimated using a variation of a particle filter (cf. (Petuchowski & Lease, 2014)). Hence, SFilter, evaluated entirely by simulation, is inconsistent with what we see in real data, such as in Figure 1. In contrast, Jung et al. (Jung et al., 2014) relax the special conditions ($c = 0$ and $\phi = 1$) by proposing a general time-series approach ($x_t = c + \phi x_{t-1} + \epsilon_t$). The principal difference is to capture and summarize the underlying dynamics of workers' label correctness more accurately.

Apart from modeling temporal dynamics behind crowd work, most prior work in crowdsourcing has focused on simple estimation of workers' performance via metrics such as accuracy and F1 (Kazai, 2011) (Smucker & Jethani, 2011). Unlike other studies, Caterette and Soboroff presented several annotator models based on Bayesian-style accuracy with various types of Beta priors (Carterette & Soboroff, 2010). Recently, Ipeirotis and Gabrilovich presented a similar type of Bayesian style accuracy with a different Beta prior in order to measure workers' performance (Ipeirotis & Gabrilovich, 2014). However, none of these studies investigated prediction of a worker's label quality.

Figure 1 shows two real examples of existing annotator models' failures in predicting worker's label correctness. The more accurate left worker (a) begins with very strong accuracy (0.8) which continually degrades over time, whereas the accuracy of the right worker (b) hovers steadily around 0.5. Suppose that a crowd worker's next label quality (y_t) is binary (correct/wrong) with respect to ground truth. While y_t oscillates over time, the existing models are not able to capture such temporal dynamics and thus prediction based on these models is almost always wrong. In particular, when a worker's labeling accuracy is greater than 0.5 (eg., average accuracy = 0.67 in Figure 1 (a)), the prediction based on the existing models are always 1 (correct) even though the actual worker's next label quality oscillates over time. A similar problem happens in Figure 1 (b) with another worker whose average accuracy is below 0.5.

Problem Setting. We begin with a binary label annotation problem in crowdsourcing. Suppose that a worker has completed n labels, and that for each label we also have ground truth available. Our task is to predict whether or not a worker's next label will be correct, as defined by agreement with ground truth. The correctness of the ith label is denoted as $y_i \in \{0, 1\}$, where 1 and 0 represent correct or not. Thus, the performance of a worker can be represented as a sequence of binary observations, $\mathbf{y} = \begin{bmatrix} y_1 & y_2 & \dots & y_n \end{bmatrix}$. For example, if a worker completed five labels and erred on the first and third respectively, then his *binary performance sequence* is encoded as $\mathbf{y} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \end{bmatrix}$. **GOLD** in Figure 1 indicates $\mathbf{y}$ of each worker, which means the binary correctness of each label.

For this problem, we propose a *Generalized Annotator Model* (GAM) that allows us to flexibly capture a wide range of workers' behaviors by incorporating features which model different aspects of this behavior. By this ability to flexibly model more aspects of worker behavior, we expect greater predictive power and an opportunity for more accurate predictions.

We generate a multi-dimensional feature vector, $x_i = \begin{bmatrix} x_{1i} & x_{2i} & \dots & x_{mi} \end{bmatrix}$ per time i and use x_i as an input of a prediction function f. Prior annotator models only consider a single generative feature x_i by a single metric, accuracy, and then use this feature as an input of simple link function y_{i+1} = *roundOff*(x_i). Instead, our proposed model incorporates a multi-dimensional feature vector x_i and uses this feature vector with a learning framework $f(x_i, y_i) = y_{i+1}$. The bottom plot of Figure 1 shows how GAM is able to track the worker's varying correctness with greater fidelity.

3. Method: Generalized time-varying Annotator Model (GAM)

In this section, we present a generalizable feature-based annotator model that incorporates various observable and latent features modeling different aspects of workers' behavior. We first examine feature generation and integration, and then discuss learning a predictive model with the generated features.

3.1. Feature Generation and Integration

A worker's behavior and annotation performance may be captured by various types of features. In this study, we generate and integrate two types of features shown in Table 1: observable and latent features. Bayesian-style features have various forms in prior work according to different Beta prior settings. Among them, we adopt $optimistic$ (a Beta prior $\alpha = 16, \beta = 1$) and $pessimistic$ (a Beta prior $\alpha = 1, \beta = 16$) annotator models from Carterette and Soboroff's study (Carterette & Soboroff, 2010). In addition, we adopt a Bayesian style accuracy from Ipeirotis and Gabrilovich's study which assumes a Beta prior ($\alpha = 0.5, \beta = 0.5$), referred to here as the $uniform$ annotator

	Feature Name	Description
Observable	Bayesian Optimistic Accuracy (BA_{opt}) (Carterette & Soboroff, 2010)	a Bayesian style accuracy with a prior $Beta$ (16,1) $BA_{opt} = (x_t + 16)/(n_t + 17)$
	Bayesian Pessimistic Accuracy (BA_{pes}) (Carterette & Soboroff, 2010)	a Bayesian style accuracy with a prior $Beta$ (1,16) $BA_{pes} = (x_t + 1)/(n_t + 17)$
	Bayesian Uniform Accuracy (BA_{uni}) (Ipeirotis & Gabrilovich, 2014)	a Bayesian style accuracy with a prior $Beta$ (0.5,0.5) $BA_{uni} = (x_t + 0.5/(n_t + 1)$
	Sample Running Accuracy (SA)	$SA_t = x_t/n_t$
	CurrentLabelQuality	a binary value indicating whether a current label is correct or wrong.
	TaskTime	time to spend in completing this label annotation task. (ms)
	AccuracyChangeDirection (ACD)	a binary value indicating the absolute difference between $SA_{t-1} - SA_t$.
	TopicChange	a binary value indicating a topic change between time $t-1$ and time t.
	NumLabels	a cumulative number of completed labels at time t.
	TopicEverSeen	a real value [0∼1] indicating the familiarity of a topic. $\frac{1}{a\ number\ of\ labels\ on\ topic\ k\ at\ time\ t}$
Latent	Asymptotic Accuracy (AA) (Jung et al., 2014)	a time-series accuracy estimated by latent time-series model proposed by Jung et al. $\frac{c}{1-\phi}$.
	ϕ (Jung et al., 2014)	a temporal correlation indicating how frequently a sequence of correct/wrong observations has changed over time.
	c (Jung et al., 2014)	a variable indicating the direction of labels between correct and wrong.

Table 1. Features of generalized annotator model (GAM). n is the number of total labels and x is the number of labels at time t.

model. In these worker models, each Beta prior characterizes each worker's annotation performance. For instance, the $optimistic$ annotator model indicates that an worker is likely to make a label in a permissive fashion, while the $pessimistic$ model tends to make more negative labels than positive label. The $uniform$ model has an equal chance of making a positive or negative label. Note that Bayesian style accuracies ($BA_{opt}, BA_{pes}, BA_{uni}$) were only used as a way of simulating labels or estimating a worker's performance in the original studies. In this study, we instead used these accuracies as a feature of estimating a worker's annotation performance as well as predicting a worker's next label correctness. Other observable features include measurable features from a sequence of labels from a worker. Among them, $TaskTime$ and $NumLabels$ are designed to capture a worker's behavioral transition over time. $TopicChange$ checks the sensitivity of a worker to topic variation over time. The $TopicEverSeen$ feature is designed to consider the effect of growing topic familiarity over time. The value is discounted by increased exposure to topic k.

Latent features are adopted from Jung et al's (Jung et al., 2014) model of temporal dynamics of worker behavior (ϕ and c). While they only used $asymptotic\ accuracy$ (AA) as an indicator of a worker's annotation performance, we integrate all three features (AA, ϕ, and c) into our generalized annotator model. Our intuition is that each feature may capture a different aspect of a worker's annotation performance and thus the integration of various features enabling greater predictive power for more accurate predictions.

3.2. Predicting Label Quality

To select a learning model, we adopt **L1-regularized logistic regression** due to several reasons. Firstly, it supports probabilistic classification as well as binary prediction by logistic function. In our problem setting, we conflate multi class labels into binary values (0 or 1), and thus logistic regression is the best fit in order to handle such a binary classification problem. In addition, a logistic regression model allows us obtain the odds ratio, defined as the ratio of the probability of correct over incorrect labels. Secondly, L1-regularized logistic regression prevents over-fitting in learning models due to either co-linearity of the covariates or high-dimensionality. The regularized regression shrinks the estimates of the regression coefficients towards zero relative to the maximum likelihood estimate. Finally, logistic regression is relatively simple and fast. In practice, one of the challenging issues to run learning algorithms is that it takes too much time to update parameters and predict output values once a new label comes. However, this model is quite efficient.

In prediction, we consider a supervised 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, and $y_i \in 0, 1$ is a class label indicating whether a worker's next label is correct (1) or wrong (0). Before fitting a model to our feature and target labels, we first normalize our features in order to ensure that normalized feature values implicitly weight all features equally in a model learning process. Logistic regression models the probability distribution of the class label y given a feature vector X as follows:

$$p(y=1|x;\theta) = \sigma(\theta^T x) = \frac{1}{1+exp(-\theta^T x)} \quad (1)$$

Here $\theta = \{\beta_0, \beta_1^T, ..., \beta_M^T\}$ are the parameters of the logistic regression model; $\sigma(\cdot)$ is the sigmoid function, defined by the second equality. The following function attempts to maximize the log-likelihood in order to fit a model to a given training data.

$$\max_{\theta}\{\sum_{i=1}^{N}[y_i(\beta_0+\beta^T x_i) - log(1+e^{\beta_0+\beta^T x_i})] - \lambda\sum_{j=1}^{M}|\beta_j|\}. \quad (2)$$

3.3. Prediction with Decision Reject Option

Our predictive model can generate two types of outputs: a binary value predicting the correctness of a worker's label (0 or 1) and a continuous value ($y_{i+1} \in [0,1]$) indicating the probability of making a correct label. While a binary predictive value (*hard prediction*) can be used as it is, a probabilistic predicted value (*soft prediction*) can be used after a transformation, such as rounding-off. For instance, if an original predicted value is 0.76, we could round this to a binary predictive value of 1.

In term of soft prediction, there exists room for improving its quality by taking account of prediction confidence. For instance, if a value of soft prediction is close to 0.5, it fundamentally indicates very low confidence. Therefore, we may avoid the risk of getting noisy predictions by adopting a *decision rejection option* (Pillai et al., 2013). In this study, we round off a probabilistic predictive value with a decision reject option as follows. If $y_{i+1} < 0.5-\delta$ or $y_{i+1} \geq 0.5+\delta$ then y_{i+1} does not need any transformation and use its original value. If $y_{i+1} \geq 0.5-\delta$ or $y_{i+1} < 0.5+\delta$ then y_{i+1} is $null$, indicating the reject of decision. δ is a parameter to control the limits of decision reject option $\in [0, 0.5]$. High δ indicates a conservative prediction which increases the range of decision rejection while sacrificing coverage. On the other hand, low δ allows prediction in a permissive manner, decreasing the threshold of decision rejection and increasing coverage.

4. Evaluation

Experimental Settings

Dataset. Data from the NIST TREC 2011 Crowdsourcing Track Task 2 is used. The dataset contains 89,624 *graded relevance judgments (2: strongly relevant, 1: relevant, 0: non-relevant)* collected from 762 workers rating the relevance of different Webpages to different search queries (Buckley et al., 2010). We conflate graded judgment labels into a binary scale (relevant / non-relevant). We processed this dataset to extract the original temporal order of the worker's labels. We include 3,275 query-document pairs which have ground truth by NIST assessors, and we exclude workers making < 20 labels to ensure stable estimation. Moreover, since the goal of our work is to predict workers' next label quality, we intentionally focus on prolific workers who will continue to do this work in the future, for whom such predictions will be useful. 54 sequential label sets are obtained, one per crowd worker. The average number of labels (i.e., sequence length) per worker is 154.

Metrics. We evaluate the performance of our prediction model with two metrics. Firstly, we measure the prediction performance with accuracy and *Mean Absolute Error* (MAE). Predicted probabilistic values (soft prediction) produced by our model are measured with MAE, indicating the absolute difference between a predicted value vs. original binary value indicating the correctness of a worker's label: $MAE = \frac{1}{n}\sum_{i=1}^{n}|pred_i - gold_i|$, where n is the number of labels by a worker. Rounded binary labels (hard prediction) are evaluated by accuracy. Secondly, accuracy is used for measuring the prediction performance of the binary probabilistic values from our prediction method. Since our extracted dataset is well-balanced in terms of a ratio between positive vs. negative labels, use of accuracy is appropriate.

Models. We evaluate our proposed Generalized Annotator Model (GAM) under various conditions of *decision reject options* with two metrics. Our initial model uses no decision reject option, setting $\delta = 0$. In order to examine the effect of *decision reject options*, we vary $\delta \in [0, 0.25]$ by 0.05 step-size. Since we have 54 workers, we build 54 different predictive models and evaluate their prediction performance and final label quality improvement.

Our model works in a sequential manner that updates the model parameter θ once a new binary observation value (correct/wrong) comes. We use each worker's first 20 binary observation values as an initial training set. For instance, suppose a worker has 50 sequential labels. We first collect a sequence of binary observation values (correct/wrong) by comparing a worker's label with a corresponding ground truth judged by NIST experts. Next, our prediction model takes the first 20 binary observation values and then predicts the 21st label's quality (correct/wrong) of this worker. Once actual 21st label comes from this worker, we measure the accuracy and MAE by comparing the label with a corresponding ground truth. For the following 29 labels we repeat the same process in a sequential manner, predicting the quality of each label one-by-one.

To learn our logistic regression model, we choose the regularization parameter λ as 0.01 after the investigation of prediction performance with varying parameter values $\{0.1, 0.01, 0.001\}$ over the initial training set of each worker. For feature normalization, we apply standard min-max normalization to the 13 features defined in Section 3.1.

Metric	GAM	TS	BA_{uni}	BA_{opt}	BA_{pes}	SA
Accuracy	0.802*	0.621	0.599	0.601	0.522	0.599
% Improvement	NA	29.1	33.9	33.4	53.6	33.9
# of Wins	NA	50	52	50	54	52
# of Ties	NA	3	1	3	0	1
# of Losses	NA	1	1	1	0	1
MAE	0.340*	0.444	0.459	0.448	0.488	0.458
% Improvement	NA	23.4	25.9	24.1	33.0	25.8
# of Wins	NA	53	53	53	54	53
# of Losses	NA	1	1	1	0	1

Table 2. Prediction performance (Accuracy and Mean Average Error) of different predictive models. *% Improvement* indicates an improvement in prediction performance between GAM vs. each baseline ($\frac{(GAM-baseline)}{baseline}$). *# of Wins* indicates the number of workers that GAM outperforms a baseline method while *# of Losses* indicates the opposite of *# of Wins*. *# of Ties* indicates the number of workers that both a method and GAM show the same prediction performance for a worker. (*) indicates that GAM prediction outperforms the other six methods with a high statistical significance ($p<0.01$).

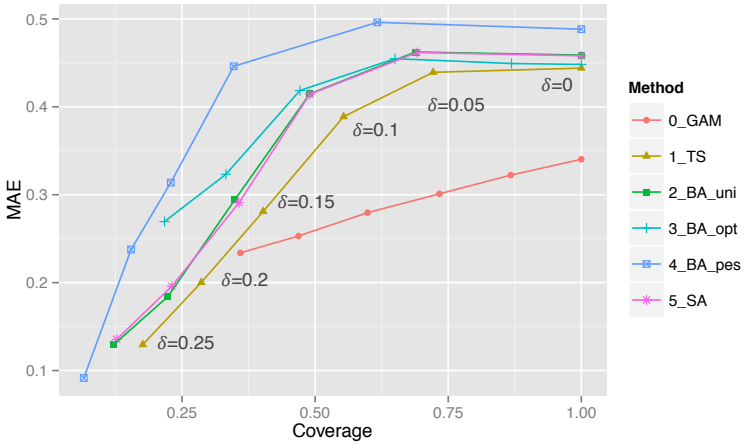

Figure 2. Prediction performance (MAE) of workers' next labels and corresponding coverage across varying decision rejection options (δ=[0-0.25] by 0.05). While the other methods show a significant decrease in coverage, under all of the given reject options, GAM shows better coverage as well as prediction performance.

Note that λ is the only model parameter we tune, and all settings of decision-reject parameter are reported in results.

As a baseline, we consider several crowd annotator models proposed by prior studies (Carterette & Soboroff, 2010) (Ipeirotis & Gabrilovich, 2014) (Jung et al., 2014) (Section 3.1). We adopt two annotator models from Carterette and Soboroff's study, *optimistic* annotator (BA_{opt}) and *pessimistic* annotator (BA_{pes}), and one annotator model of Bayesian accuracy (BA_{uni}) used in Ipeirotis and Gabrilovich's study (see Table 1). In addition, we test the performance of a time-series model (TS) proposed by Jung et al (Jung et al., 2014) and sample running accuracy (SA) as defined by Table 1. All of the baseline methods predict the binary correctness of the next label y_{i+1} by rounding off the worker's estimated accuracy at time i. *Decision reject options* are equally applied to all of the baseline methods.

4.1. Experiment 1 (RQ1): Prediction Performance Improvement

To answer our first research question, we compare the overall prediction performance (Accuracy, MAE) of GAM with the baseline models across 54 crowd workers. **Table 2** shows that GAM prediction performance outperforms all of the baseline methods across 50-54 workers in accuracy and 53-54 workers in MAE. GAM improves the prediction accuracy (hard label) and MAE (soft label) by 26-36% on average. GAM prediction errs for only one worker vs. the baselines. However, even for this worker, GAM only made one or two more prediction errors in comparison to the other baselines.

Furthermore, we examine the effects of *decision reject options* on GAM prediction. Figure 2 demonstrates that the baseline models show sharp decline of coverage in prediction in order to significantly improve their prediction accuracies. However, the coverage of GAM prediction only gently decreases; even with the second strongest reject option ($\delta = 0.2$), it still covers almost the half of prediction. In sum, GAM prediction not only outperforms the baseline models in terms of prediction accuracy, but it also shows less sensitivity to the increase of the decision reject option.

4.2. Experiment 2 (RQ2): Impact on label quality and cost

Our next experiment is to examine quality effects on crowd labels via the proposed prediction model. We conduct an experiment based on task routing. For instance, if the prediction of a worker's next label indicates that the worker is expected to be correct, we route the given example to this worker and measure actual label quality against ground truth labeled by NIST. From our dataset, we only use 826 examples that have more than three crowd labels per example. Since the average number of workers per example is about 3.7, we test the cost saving effect with varying three task routing scenarios (*Number of Workers* = $\{1, 2, 3\}$). Label quality is measured with accuracy, and a paired t-test is conducted to check whether quality improvement is statistically significant.

Table 3 shows the results of label quality via predictive model-based task routing. GAM substantially outperforms the other baselines across three task routing cases. The improvement of final label quality grows with the increase of the number of workers per example (*Number of Judges*) from 29-32% to 36-47%. Notice that GAM with only two routed workers achieves 29% quality improvement. Moreover, GAM provides high-quality crowd labels (accuracy > 0.8) with only 54% = ($\frac{2}{3.7}$) of the original assessment cost. In contrast, we see that task routing with baselines

	Prediction Models for Task routing							No Routing
Number of Workers	GAM	TS	BA_{uni}	BA_{opt}	BA_{pes}	SA	**Random**	**All labels**
1	0.786*	0.604	0.578	0.582	0.558	0.569	0.556	0.595
% Improvement	NA	30.1	36.0	35.1	40.9	38.1	41.4	
2	0.816**	0.617	0.592	0.595	0.574	0.582	0.572	
% Improvement	NA	32.3	37.8	37.1	42.2	40.2	42.7	
3	0.880*	0.647	0.608	0.623	0.598	0.608	0.581	
% Improvement	NA	36.0	44.7	41.3	47.2	44.7	51.5	

Table 3. Accuracy of labels via predictive models. *Number of Workers* indicates the number of workers per example. When the *Number of workers* > 1, majority voting is used for label aggregation. Accuracy is measured against ground truth. *% Improvement* indicates an improvement in label accuracy between GAM vs. each baseline ($\frac{(GAM-baseline)}{baseline}$). The average number of workers per example is 3.7. (*) indicates that GAM prediction outperforms the other six methods with high statistical significance (p<0.01).

alone (BA_{uni},BA_{pes},SA) may not be any better than random assignment.

4.3. Experiment 3 (RQ3): Impact of limited supervision

In practice, it may be challenging to have gold labels, such as NIST expert labels, to judge the binary correctness of each crowd worker's label. To relieve this concern, our last experiment investigates to what extent our prediction model is influenced by limiting the number of training labels.

While our earlier experiments (Experiment 1-2) assumed the existence of gold labels to judge the binary correctness of all crowd worker labels, we now limit the number of training labels by assuming that we only have a small number of gold labels. In this setting, for instance, given 50 labels by a crowd worker, our prediction model is trained with only the 10 initial training labels (binary correctness) and then predicts the binary correctness of the crowd worker's remaining 40 labels. We investigate the effect of the limited number (k) of training labels on the prediction accuracy of our prediction model by changing k from 11 to 30.

Figure 3 shows how our prediction model (GAM) and the other prediction models perform under varying the number (k) of training labels. Note that GAM shows significant improvement of prediction accuracy as the size of k increase. When k is relatively small (between 10-15), its prediction accuracy ranges from 0.6 to 0.66. As training labels are additionally provided, its prediction performance tends to improve up to 0.70. On the contrary, all other baseline models except BA_{PES} do not show noticeable improvement of prediction accuracy in spite of the increase of training labels. In terms of BA_{PES}, prediction accuracy is still very low even though it steadily increases it prediction performance with the increase of training labels. In sum, this experiment demonstrates that the prediction performance of GAM improves steadily with the increase of training labels and its prediction accuracy is significantly higher than the other baseline models. In addition, its prediction accuracy is reasonably good (0.7) despite using the limited number of training labels.

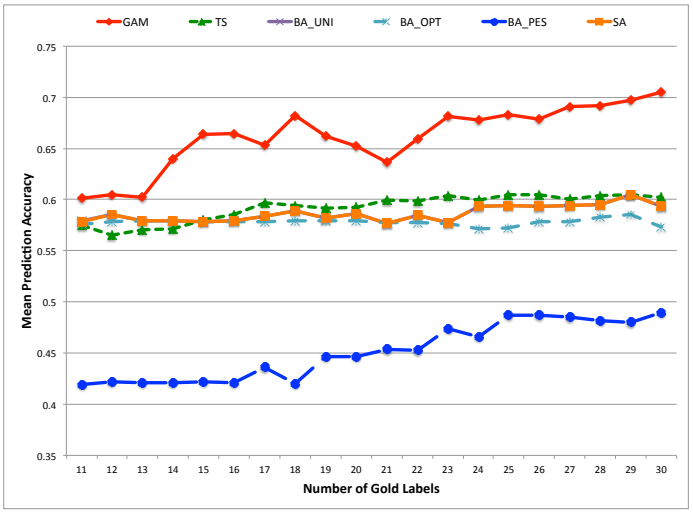

Figure 3. Impact of limited supervision on prediction performance. The X-axis indicates the number of gold labels which are used for learning each predictive model. The Y-axis indicates the mean prediction accuracy across 54 workers.

Table 4 shows the comparison of prediction performance between GAM vs. the other prediction models when using only 30 training labels. Since we already confirmed that MAE shows a similar pattern to accuracy, we only report prediction accuracy of each model in this experiment. In comparison to Table 2, overall prediction accuracy decreases due to the limited number of training labels. However, GAM outperforms the other baseline models by 15.3-44.0%. With regards to win/loss, GAM still outperforms the baseline models by 83.3-91.1%.

Figure 4 plots for each worker the difference in prediction accuracy between GAM vs. the best possible baseline for that worker (i.e., oracle selection). We choose to use the best possible baseline for each worker to provide the most conservative analysis of lower-bound, rela-

Metric	GAM	TS	BA_{uni}	BA_{opt}	BA_{pes}	SA
Accuracy	0.705*	0.611	0.593	0.573	0.490	0.569
% Improvement	NA	15.3	18.9	23.1	44.0	23.9
# of Wins	NA	40	41	41	46	41
# of Ties	NA	6	9	8	3	9
# of Losses	NA	8	4	5	5	4
Winning Ratio (%)	NA	83.3	91.1	89.1	90.2	91.1

Table 4. Prediction accuracy of different predictive models with limited supervision (k=30). *% Improvement* indicates an improvement in prediction performance between GAM vs. each baseline ($\frac{(GAM-baseline)}{baseline}$). *# of Wins* indicates the number of workers that GAM outperforms a baseline method while *# of Losses* indicates the opposite of *# of Wins*. *# of Ties* indicates the number of workers that both a method and GAM show the same prediction performance for a worker. Winning Ratio (%) means a number of wins over the total number of predictions ($\frac{(NumberofWins)}{NumberofWins+NumberofLosses}$). (*) indicates that GAM prediction outperforms the other five methods with a high statistical significance ($p<0.01$).

tive improvement provided by GAM. While TS is selected in the most workers (70%, 38 workers), BA_{UNI}, SA, and BA_{OPT} are selected in the rest of the cases. We observe the similar pattern in Figure **??** that GAM significantly improves prediction accuracy for workers whose labeling accuracy ranges from 0.4-0.6. In other words, despite limited supervision, our prediction models shows significant improvement of prediction accuracy for noisy workers while the other baseline models show relatively poor performance in predicting the correctness of noisy workers' next labels.

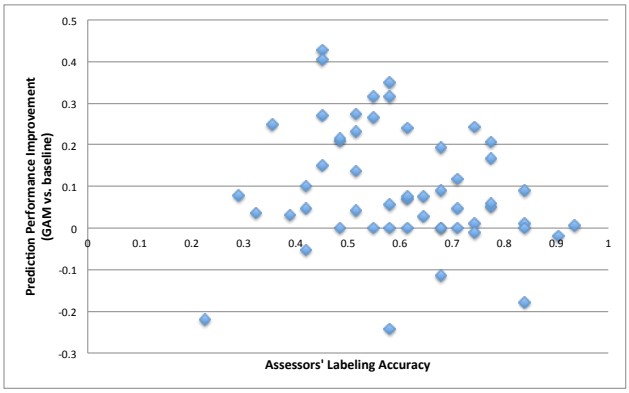

Figure 4. Relative difference of prediction accuracy (GAM-baseline) vs. workers' labeling accuracy under limited supervision (30 gold labels). The best possible baseline is selected per worker. The x-axis indicates a worker's labeling accuracy. The y-axis indicates the relative difference of prediction accuracy between GAM vs. the baseline.

To sum up, our last experiment demonstrates that GAM still performs significantly better than the other baseline models in terms of predicting a crowd worker's next label quality given limited number of training labels. In particular, GAM allows us to predict noisy workers' label quality more accurately than the other baselines. This result demonstrates that our prediction model can indeed be utilized in practice with only a small number of gold labels.

5. Conclusion and Future Work

Despite recent efforts of quality improvement in crowdsourced labels, prior work in crowd worker modeling cannot adequately predict a worker's next label quality since it simply measures worker performance via a single generative model without considering temporal effects among labels. We present a general discriminative learning framework for integrating arbitrary and diverse evidence for temporal modeling and prediction of crowd work accuracy. Our experiments demonstrate that the proposed model improves prediction performance by 26-36% as well as crowdsourced labels quality by 29-47% at 17-45% lower cost. Furthermore, we confirm that our model still performs significantly better than the other baselines under limited supervision with modest performance degradation.

As a next step, we plan to relax our restrictive assumption of the existence of gold labels to judge the correctness of a worker's labels. Beyond that, we plan to further investigate how to use this model for different applications of quality assurance in crowdsourcing, such as weighted label aggregation and spam worker filtering.

Acknowledgments

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

Alonso, Omar, Rose, Daniel E., and Stewart, Benjamin. Crowdsourcing for relevance evaluation. *ACM SIGIR Forum*, 42(2):9–15, 2008.

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

Buckley, Chris, Lease, Matthew, and Smucker, Mark D. Overview of the TREC 2010 Relevance Feedback Track (Notebook). In *Proceedings of TREC*, 2010.

Carterette, Ben and Soboroff, Ian. The effect of assessor error on ir system evaluation. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pp. 539–546, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0153-4. doi: 10.1145/1835449.1835540. URL http://doi.acm.org.ezproxy.lib.utexas.edu/10.1145/1835449.1835540.

Donmez, Pinar, Carbonell, Jaime, and Schneider, Jeff. A probabilistic framework to learn from multiple annotators with time-varying accuracy. In *SIAM International Conference on Data Mining (SDM)*, pp. 826–837, 2010. URL http://www.cs.cmu.edu/afs/cs/Web/People/pinard/Papers/280_Donmez.pdf.

Ipeirotis, Panagiotis G and Gabrilovich, Evgeniy. Quizz: targeted crowdsourcing with a billion (potential) users. In *Proceedings of the 23rd international conference on World wide web*, pp. 143–154. International World Wide Web Conferences Steering Committee, 2014.

Jung, Hyun Joon and Lease, Matthew. A Discriminative Approach to Predicting Assessor Accuracy. In *Proceedings of the European Conference on Information Retrieval (ECIR)*, 2015. URL ../papers/ecir2015_hjung.pdf. Received *Samsung Human-Tech Paper Award: Silver Prize in Computer Science.*

Jung, Hyun Joon, Park, Yubin, and Lease, Matthew. Predicting Next Label Quality: A Time-Series Model of Crowdwork. In *Proceedings of the 2nd AAAI Conference on Human Computation (HCOMP)*, 2014. To appear. 9 pages.

Kazai, Gabriella. In search of quality in crowdsourcing for search engine evaluation. In *ECIR'11 Proceedings of the 30th European conference on Advances in information retrieval*, pp. 165–176, 2011. URL http://www.springerlink.com/index/U33373T17P56HR2L.pdf.

Kazai, Gabriella, Kamps, J, and Milic-Frayling, N. The face of quality in crowdsourcing relevance labels: demographics, personality and labeling Accuracy. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pp. 2583–2586, 2012. ISBN 9781450311564. URL http://staff.science.uva.nl/~kamps/readme/publications/2012/kaza:face12.pdf.

Law, Edith, Bennett, PN, and Horvitz, Eric. The effects of choice in routing relevance judgments. In *Proceedings of the 34th ACM SIGIR conference on Research and development in Information*, pp. 7–8, 2011. URL http://dl.acm.org/citation.cfm?id=2010082.

Lease, Matthew and Kazai, Gabriella. Overview of the TREC 2011 Crowdsourcing Track (Conference Notebook). In *20th Text Retrieval Conference (TREC)*, 2011. Final proceedings version forthcoming.

Petuchowski, Ethan and Lease, Matthew. TurKPF: TurKontrol as a Particle Filter. Technical report, University of Texas at Austin, April 2014. arXiv:1404.5078.

Pillai, Ignazio, Fumera, Giorgio, and Roli, Fabio. Multi-label classification with a reject option. *Pattern Recognition*, 46(8):2256 – 2266, 2013. ISSN 0031-3203.

Raykar, VC and Yu, S. Eliminating spammers and ranking annotators for crowdsourced labeling tasks. *Journal of Machine Learning Research*, 13:491–518, 2012. URL http://dl.acm.org/ft_gateway.cfm?id=2188401&type=pdf.

Rzeszotarski, JM and Kittur, Aniket. Instrumenting the crowd: Using implicit behavioral measures to predict task performance. In *Proceedings of the 24th annual ACM symposium on User interface software and technology (UIST)*, 2011. ISBN 9781450307161. URL http://dl.acm.org/citation.cfm?id=2047199.

Sheng, Victor S., Provost, Foster, and Ipeirotis, Panagiotis G. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proc. ACM KDD*, pp. 614–622, 2008.

Smucker, Mark D and Jethani, Chandra Prakash. Measuring assessor accuracy: a comparison of NIST assessors and user study participants. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pp. 1231–1232, 2011. ISBN 9781450307574.

Venanzi, Matteo, Guiver, John, Kazai, Gabriella, Kohli, Pushmeet, and Shokouhi, Milad. Community-based Bayesian Aggregation Models for Crowdsourcing. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pp. 155–164, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2744-2. doi: 10.1145/2566486.2567989. URL http://doi.acm.org/10.1145/2566486.2567989.

Vuurens, Jeroen B.P. and de Vries, Arjen P. Obtaining High-Quality Relevance Judgments Using Crowdsourcing. *IEEE Internet Computing*, 16(5):20–27, September 2012. ISSN 1089-7801. doi: 10.1109/MIC.2012.71. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6216343.

Yi, Jinfeng, Jin, Rong, Jain, Shaili, and Jain, Anil K. Inferring Users' Preferences from Crowdsourced Pairwise Comparisons: A Matrix Completion Approach. In *1st AAAI Conference on Human Computation (HCOMP)*, 2013.

Yuen, Man-Ching, King, Irwin, and Leung, Kwong-Sak. Task recommendation in crowdsourcing systems. In *Proceedings of the First International Workshop on Crowdsourcing and Data Mining*, CrowdKDD '12, pp. 22–26, 2012.