Predicting Next Label Quality: A Time-Series Model of Crowdwork

Hyun Joon Jung
School of Information
University of Texas at Austin
hyunJoon@utexas.edu

Yubin Park
Dept. of Computer & Electrical Engineering
University of Texas at Austin
yubin.park@utexas.edu

Matthew Lease
School of Information
University of Texas at Austin
ml@utexas.edu

Abstract

While temporal behavioral patterns can be discerned to underlie real crowd work, prior studies have typically modeled worker performance under a simplified i.i.d. assumption. To better model such temporal worker behavior, we propose a time-series label prediction model for crowd work. This latent variable model captures and summarizes past worker behavior, enabling us to better predict the quality of each worker's next label. Given inherent uncertainty in prediction, we also investigate a *decision reject option* to balance the tradeoff between prediction accuracy vs. coverage. Results show our model improves accuracy of both label prediction on real crowd worker data, as well as data quality overall.

Keywords: *task routing, recommendation, time series*

Introduction

For online crowd work (Kittur et al. 2013), effective task recommendation and routing have potential to significantly improve the quality of data collected and worker experience by better matching workers to available work (Law, Bennett, and Horvitz 2011; Li et al. 2014). Whereas preference-based recommendation models varying worker interest for different task types, performance-based recommendation models varying worker accuracy as a function of task type (*macro-level* worker-task matching) or specific examples (*micro* quality prediction). Prior work in performance-based recommendation has typically modeled behavior of crowd workers as independent and identically distributed (i.i.d.) over time (Yuen, King, and Leung 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 and Soboroff 2010). Regardless of cause, temporal effects are clearly evident.

The closest prior work we are aware of on temporal modeling of crowd work, by Donmez et. al. (2010), assumes that workers are weak learners who behave according to simple latent dynamics $x_t = x_{t-1} + \epsilon_t$. This approach, based entirely on simulation, assumes a uniform offset and temporal correlation for the underlying dynamics, inconsistent with what we see in real data, such as in Figure 1.

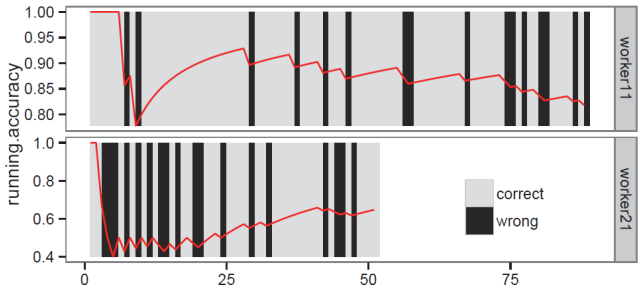

Figure 1: The work quality of two crowd workers is seen to vary over time. Gray area indicates correct responses while black stripes denote errors. The *running accuracy* of each (i.e., empirical accuracy up to a given response) is plotted in red. The top worker's accuracy is seen to decrease over time, while the lower worker's accuracy improves. Their temporal error patterns also differ: the top worker's errors become more frequent while the lower worker's become less so.

To more faithfully model such temporal behavior, we present a time series-based label prediction model for crowd workers' behavioral patterns. This categorical time series model uses a temporally-correlated latent variable which captures and summarizes the past behaviors of a worker, enabling us to better predict the quality of the next label. To efficiently estimate model parameters, we adapt a recently developed technique (Park, Carvalho, and Ghosh 2014) to our crowdsourcing problem context. Given inherent uncertainty in prediction, we also investigate a *decision reject option* to balance the tradeoff between prediction accuracy vs. coverage (Bartlett and Wegkamp 2008; Nadeem, Zucker, and Hanczar 2010).

We evaluate our time-series model on real worker data from the NIST TREC Crowdsourcing Track (sites.google.com/site/treccrowd/). Results show our model enables us to better predict workers' next labels. The *decision reject option* enables further accuracy improvement by sacrificing coverage, providing a tuning parameter for aggressive vs. conservative prediction given model confidence. Additional simulation experiments show overall quality improvements

achieved. We investigate the following research questions:

RQ1: Application of time-series modeling. *How can we apply and interpret time-series models of crowd work?*

RQ2: Label prediction via a time series model. *How accurately does time-series prediction work in our context?*

RQ3: Use of *decision reject option* for managing uncertainty *How effectively does the decision reject option let us tradeoff our model's prediction accuracy vs. coverage?*

RQ4: Label quality improvement. To what extent can time-series modeling yield overall improvement in quality of crowdsourced labels?

Problem

On crowd work platforms such as Amazon's Mechanical Turk (MTurk), a task is usually self-selected by a worker. Relatively few studies have investigated task routing for micro-tasks, though work exists with other forms of crowdsourcing, such as Wikipedia (Cosley et al. 2007). Kamar et al. (2012) studied the cooperative refinement and task routing among on-line agents. In addition, both Kamar et al. and Dai et al. (2010) developed methods to predict the accuracy of the next label, but did not model workers' individual temporal profiles in making these predictions. Bernstein et al. (2012) investigated task routing in terms of real-time crowdsourcing. These studies do not address finding strong candidates for a particular task from the requester's viewpoint. Work on task markets seeks to chain together different worker competencies (Shahaf and Horvitz 2010).

SFilter, proposed by Donmez et. al. (2010), is a Bayesian time series model that captures crowd workers' dynamically varying performance. The authors do not learn the parameters for the latent variable dynamics, but as mentioned earlier, assumed that 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 and Lease 2014). This assumption does not seem to hold on the real crowd data we have observed, as evidenced in Figure 1.

In our paper, we attempt to relax special conditions ($c = 0$ and $\phi = 1$) by proposing a general time series model ($x_t = c + \phi x_{t-1} + \epsilon_t$). The principal difference is to capture and summarize the underlying dynamics of workers' labeling more efficiently and accurately. Our goal is to predict the next label of a crowd worker by estimating the latent variables governing the performance of the crowd worker. In addition, we would like to use the latent variables to better analyze the varying temporal performance of crowd workers. We formally define our problem as follows:

Problem. *Given an individual's performance history, 1) estimate the probability of correct labeling for the next task instance, and 2) provide meaningful summary statistics for the behavioral pattern of the worker.*

We begin with a binary label annotation problem in crowdsourcing. The extension to multiple categorization is straightforward by changing our temporal prediction model, especially the link function. We first discuss the theoretical backgrounds of our approach and present our model in the next Section. We aim to predict the next label of each worker, using this information to identify the best workers to which examples should be routed for labeling.

Method: Latent Autoregressive Model

Suppose that a worker has completed n task instances. The correctness of the ith instance 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, $\mathbf{y} = [y_1 \quad y_2 \quad \dots \quad y_n]$. For example, if a worker completed three task instances and erred on the first only, then his *binary performance sequence* is encoded as $\mathbf{y} = [0 \quad 1 \quad 1]$.

Assume that we have two workers, Alice and Bob, who have each labeled 10 instances with performance as follows:

$$\mathbf{y}_{\text{Alice}} = [1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0]$$
$$\mathbf{y}_{\text{Bob}} = [0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1]$$

While both achieve 50% accuracy with respect to the ground truth, they exhibit quite different temporal profiles. Alice provides incorrect labels immediately after she marks correct labels. On the other hand, Bob shows a poor performance in the beginning, but he correctly labeled the last four tasks in a row. We must go beyond measuring accuracy to capture such temporal variation across workers.

Several statistical models can capture this kind of temporal variation. In a broad sense, such models fall into two classes depending on the use of latent variables: fully-observed vs. latent variable models. Fully-observed models include Mixture Transition Distribution Model (MTDM) (Raftery 1985), Markovian regression model (Kaufmann 1987; Zeger, Liang, and Albert 1988), and Discrete Autoregressive Moving Average model (DARMA) (Jacobs and Lewis 1983). Latent variable models have been successfully demonstrated in various applications such as decoding algorithms (Viterbi 1967) and speech recognition (Juang and Rabiner 1991). Such latent variable models can be further grouped into two sub-categories based on the representation of latent variables: Hidden Markov Models (HMM) (Zucchini and MacDonald 2009) use discrete latent variables, whereas State-Space Models (SSM) (Zhen and Basawa 2009) adopt continuous latent variables.

We adopt a latent autoregressive (LAR) model for categorical time series. The model is a state-space model for a categorical time series that has been less popular than HMM and SSM. This is partly because such continuous latent variables are notoriously difficult to reconstruct from categorical observations. However, the use of the LAR process provides two substantial advantages: *interpretability* and *extensibility*. Indeed, the AR process has a rich history with parsimonious theoretical results (Canova and Cicarelli 2013; Litterman 1984). The interpretation on stationarity and spectral analyses (Burg 1967) can be smoothly applied to the latent AR processes. Moreover, the latent VAR process can be easily extended to cover variants of the AR models such as ARMA (Box, Jenkins, and Reinsel 1994), Autoregressive

Conditional Heteroskedasticity (ARCH) (Engle 1982), and Generalized ARCH (GARCH) processes.

We hypothesize a latent factor x_t that governs the worker performance. This latent factor evolves over time depending on the previous value in the sequence. The sequence dynamics of the latent factors is described by a set of parameters $\boldsymbol{\theta} = \{c, \phi\}$. In essence, our model is described as follows:

$$\text{(Latent AR)} \quad x_t = c + \phi x_{t-1} + \varepsilon_t \tag{1}$$

$$\text{(Observation)} \quad p(y_t = 1) = \text{logit}^{-1}(x_t) \tag{2}$$

$$\text{(Noise model)} \quad \varepsilon_t \sim \text{Normal}(0, \sigma^2) \tag{3}$$

where $\mathbf{y} = [y_t]_{t=1}^{T}$ with $y_t \in \{0, 1\}$, and $p(y_t = 1)$ indicates the probability that $y_t = 1$.

Returning to our Alice and Bob example, we show that this LAR model captures the illustrated temporal patterns. For Alice's case, let us take $c = 0.1$, $\phi = -0.9$, and $x_0 = -1$ (initial latent state). Ignoring the effect of noise, the latent variables propagate as follows:

$$\begin{aligned} x_1 &= c + \phi x_0 = 0.1 - 0.9 \times -1 = 1 \\ x_2 &= c + \phi x_1 = 0.1 - 0.9 \times 1 = -0.8 \\ x_3 &= c + \phi x_2 = 0.1 - 0.9 \times -0.8 = 0.82 \\ &\vdots \end{aligned}$$

where the sequence of latent variables oscillate. As a result, the probability of correct labeling also oscillates over time. On the other hand, for Bob's case, let us assume that the parameters are $c = 0.1$, $\phi = 0.9$, and $x_0 = -1$. The temporal sequence of the latent factors are given as:

$$\begin{aligned} x_1 &= c + \phi x_0 = 0.1 + 0.9 \times -1 = -0.5 \\ x_2 &= c + \phi x_1 = 0.1 + 0.9 \times -0.5 = -0.35 \\ x_3 &= c + \phi x_2 = 0.1 + 0.9 \times -0.35 = -0.215 \\ &\vdots \end{aligned}$$

As can be seen, the probability of correct labeling improves over time. To estimate maximum likelihood parameters, we use a method known as *Low-resolution augmented Asymptotic Mean Regularized Expectation Maximization* (LAMORE) (Park, Carvalho, and Ghosh 2014).

Estimating the parameters from a categorical sequence involves several challenges. First, unlike continuous time series, categorical time series contain only finite bits of information. Categorical outputs can be viewed as lossy-compression from an information theoretic perspective, thus the reconstruction of the continuous latent variables suffers from a low signal-to-noise ratio. Furthermore, this noisy reconstruction increases the uncertainty of the estimated parameters, especially in the alternating minimization framework. As a result, classical alternating minimization techniques such as Expectation-Maximization become susceptible to various factors, including noisy reconstruction and multiple local optima of the log-likelihood function. LAMORE combines Method of Moments and Monte-Carlo Expectation Maximization algorithms to stabilize parameter estimation. The method can be extended to general categorical time series, e.g. tertiary categorical time series.

Adaptation to Crowdsourcing

We next discuss how to use our time-series framework in the context of crowdsourcing [RQ1]. Prior to applying this framework, we first discuss the semantics of our time series model in crowdsourcing. We then present our label prediction algorithm based on this understanding of the semantics.

Interpretation of Parameters

Our time series framework takes a sequence of observations as input and generates four types of output values: latent variables ϕ, c, x_t and an observable variable $y = logit^{-1}(x_t)$. As input, we use the worker's *binary performance sequence*, as illustrated by the earlier Alice and Bob examples from the previous section.

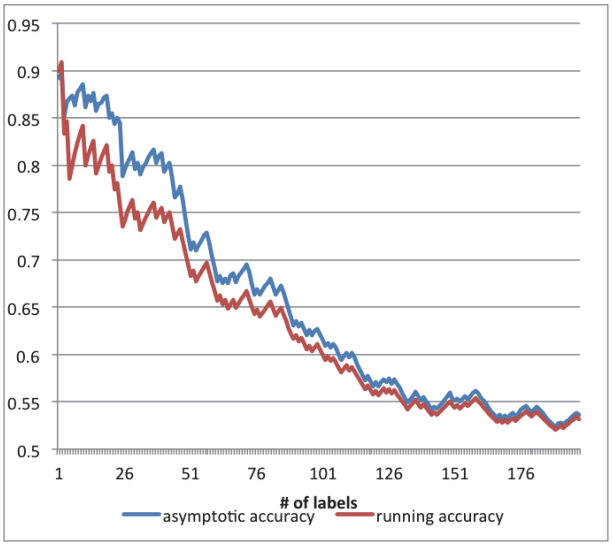

Figure 2: Relation over time between asymptotic accuracy and running accuracy (RA), where RA= $\frac{\text{\# correct labels}}{\text{\# submitted labels}}$.

Latent Variable (x_t). The interpretation of x_t has an important meaning with regard to the analysis of workers' performance. First, x_t indicates the probability of making a correct label at a time point t. In our model, a link function, $logit^{-1}(x_t)$, transforms this probability to a soft label representing the polarity of a worker's next label. With regard to task routing, this soft label is used as a criteria to judge an optimal candidate. If a soft label, $logit^{-1}(x_t)$, is close to 0 or 1, it indicates that the next label of this worker is likely to be confident. On the contrary, a label around 0.5 suggests that the confidence of the worker's next value is relatively low since the polarity of the label is weak.

Second, the dynamics of x_t is an autoregressive process with one lagged variable, AR(1). If the absolute value of the temporal correlation parameter ϕ is less than 1 i.e. $|\phi| < 1$, the underlying AR(1) process is a stationary ergodic process. The asymptotic theory of autoregressive process provides that the asymptotic mean of x_t is given as $x_\infty = \frac{c}{1-\phi}$. This can be obtained by solving $E[x_t] = E[c] + E[\phi x_{t-1}] + E[\epsilon_t] = c + \phi E[x_t]$. Since y_t is fully determined by x_t in our

model, we can extend the concept of the asymptotic mean to "asymptotic accuracy" which is defined as follows:

$$y_\infty = \text{logit}^{-1}(x_\infty) = \frac{\exp(x_\infty)}{1+\exp(x_\infty)} \quad (4)$$

Provided modeling assumptions are met, the estimated asymptotic accuracy should converge to the sample accuracy (i.e. ergodicity). **Figure 2** empirically demonstrates the convergence of these two values over time where the data comes from a randomly selected worker. This suggests that our modeling assumptions fit well to the actual data.

Temporal Correlation (ϕ). Temporal correlation ϕ indicates how frequently a sequence of correct/wrong observations has changed over time. A worker having $\phi = -0.8$ tends to follow a temporal pattern of regularly alternating between the correct and the wrong. On the other hand, another worker having $\phi = 0.8$ tends to follow a consistent pattern without a frequent switching irrespective to the correct or the wrong. Between these extremes, $\phi \approx 0$ indicates no temporal dependencies between sequential labels; a worker of having $\phi \approx 0$ does not show any regular temporal pattern. In sum, ϕ helps to characterize a worker's behavioral pattern and understand its underlying dynamics.

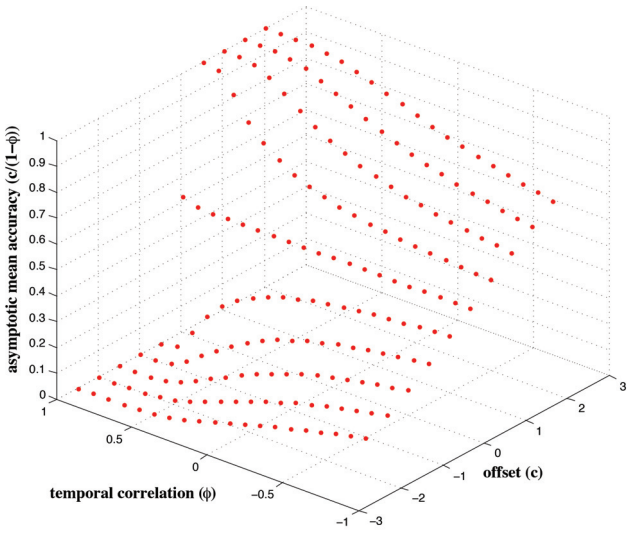

Figure 3: Relation between c and ϕ vs. asymptotic accuracy.

Offset (c). The sign of offset c navigates the direction between correct and wrong. For example, if c is positive, at each time, the latent variable of a worker is drifted toward the positive direction, which implies better a correct rate. On the other hand, if c is negative, the latent variable is drifted toward the negative direction, implying that the performance of a crowdworker will degrades over time. The size of offset c combined with ϕ determines the asymptotic accuracy of a crowdworker. **Figure 3** shows the relationship between c and ϕ vs. asymptotic accuracy $logit^{-1}(\frac{c}{1-\phi})$. When offset c is positive, the higher ϕ indicates the higher accuracy. On the contrary, the lower ϕ indicates the higher accuracy when offset c is negative. This suggests that a worker of a higher temporal correlation ϕ shows the extreme polarity with regard to her accuracy. In the mean time a worker of a low ϕ value rather shows relatively concentrated around 0.5 indicating less confidence in a worker's label.

Prediction with Reject Option

For a real application of a time series model to crowdsourcing, we first consider workers' label predictions over time. The output of the proposed time series model can be easily applied toward label prediction as follows. A value of $logit^{-1}(x_t)$ can be used as a probabilistic label (*soft label*) indicating the strength of one direction, positive or negative. For label generation, we may use this value in two ways. First, it is straightforward to use a given soft label without any transformation. Second, we generate a *hard label* based on the value of a given soft label. For instance, in terms of predicting a binary label, if a predicted soft label is 0.76, we then generate a binary label of 1 since the value of the given soft label is greater than 0.5.

In terms of label prediction, there exists a room for improving the quality of label prediction by taking account of prediction confidence. For instance, if a soft label is close to 0.5, it fundamentally indicates very low confidence in terms of the polarity. Therefore, we may avoid the risk of getting noisy predictions by adopting a decision rejection option (Pillai, Fumera, and Roli 2013). In this study, prediction with decision a reject option is defined as follows.

$$l(x_t) = \begin{cases} logit^{-1}(x_t) & \text{if } x < 0.5-\delta \quad \text{or } x \geq 0.5+\delta \\ \text{null} & \text{if } x \leq 0.5-\delta \text{ and } x < 0.5+\delta \end{cases}$$

where δ is a parameter to control the limits of decision reject option, and $\delta \in [0, 0.5]$. High δ indicates a conservative label prediction which increases the range of decision rejection while sacrificing coverage. On the other hand, low δ allows label prediction in a permissive manner, decreasing the threshold of decision rejection and increasing coverage.

Evaluation

In this section, we describe the experimental evaluation and provide our observations. We have tested the proposed time-series label prediction model under various conditions of decision reject options with a real crowdsourcing dataset.

Experimental Settings

Dataset. We use a subset of a public dataset created for the NIST TREC crowdsourcing Track 2011 Task 2. The dataset contains binary *relevance judgments* from workers rating the relevance of different Webpages to different search queries (Buckley, Lease, and Smucker 2010; Grady and Lease 2010). We processed this dataset to extract the original order of the workers' labels. We include only examples which have ground truth labels, and we exclude workers making < 20 judgments to ensure stable estimation. Moreover, since the goal of our work is to be able to route work to specific workers, it is only worth modeling a given worker's behavior if we believe that worker will continue to do more work in the future, as suggested by their having already performed some minimal amount of work. We extract

73 sequential label sets, one per crowd worker. The average number of labels (i.e., sequence length) per worker is 163. The standard deviation is large (259) due to three workers having each submitted over 600 relevance judgments.

Models. We evaluate our proposed time series model (TS-prediction model) under various conditions of decision reject options. 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.

For label prediction, we use the first 10 observed labels for training and measure the performance of prediction with the rest of sequential labels. For instance, if a worker has 50 sequential labels, our prediction model takes the first 10 correct/wrong observation from the labels and then predicts the following 40 labels in a sequential manner. The prediction algorithm generates only the next label since we focus on only short-term prediction in this study.

As a baseline, we compute a worker's *running accuracy* at time t (RA_t) as her observed accuracy up to time t, then use this value as the probability of the worker's next label being correct. While running accuracy eventually converges to asymptotic accuracy, this baseline cannot capture shorter-term dynamics of workers' behavioral patterns.

Metrics. We evaluate with two types of predicted labels. Firstly, we use probabilistic labels (soft labels) produced by our time series prediction model. Secondly, we also report evaluation with rounded binary labels (hard labels). We compute the following evaluation metrics, where tp denotes the number of true positive classifications, fp the false positives, tn the true negatives, and fn the false negatives:

$$\text{Precision (P)} = \frac{tp}{(tp+fp)} \qquad \text{Accuracy} = \frac{tp+tn}{n}$$

$$\text{Recall (R)} = \frac{tp}{(tp+fn)} \qquad \text{F1} = \frac{2PR}{P+R}$$

$$\text{LogLoss} = -\frac{1}{n}\sum_{i=1}^{n} log\, q_i(\text{IsRelevant?}(doc_i))$$

Computing these metrics for **Table 1** examples would yield:

Instance	Prediction	Truth	tp	fn	fp	tn
1	0.7	1	0.7	0.3		
2	0.6	0			0.6	0.4
3	0.2	1	0.2	0.8		
4	0.1	0			0.1	0.9

Table 1: Simple examples of predictions made on four instances, given ground truth labels indicating relevance (1) or non-relevance (0), and resulting tp, fn, fp, and tn values.

$$\text{Precision} = \frac{0.9}{0.9+0.7} \approx 0.56 \qquad \text{Accuracy} = \frac{0.9+1.3}{4}$$

$$\text{Recall} = \frac{0.9}{0.9+1.1} = 0.45 \qquad \text{F1} = \frac{2*0.45*0.56}{0.45+0.56}$$

$$\text{LogLoss} = -\frac{log(0.7)+log(0.4)+log(0.2)+log(0.9)}{4}$$

While accuracy is the simplest measure of label quality, it is less meaningful when the underlying class distribution is skewed. In such cases, the harmonic mean of precision and recall (F1) is typically more appropriate if a single measure of quality is required. In addition, we report average LogLoss with respect to each worker's predicted labels. The log function emphasizes penalties for being both confident and wrong. In the worst case, 100% confidence in a wrong label will add negative infinite error.

Experiment 1 (RQ2): Prediction without Rejection

How accurately does our time series prediction model infer the next label of workers? We first measure prediction performance of our proposed model (TS-based prediction) and running accuracy-based prediction model (RA-based prediction) over actual workers' correct/wrong observations without considering any decision reject option. **Figure 4** shows the difference between two models across 73 workers with respect to three metrics. In most cases, TS-based prediction outperforms the performance of RA-based prediction. In particular, Figure 4 (c) indicates that the proposed model makes better predictions for all but 7 workers. In the case of hard labels, overall patterns across 73 workers are identical to this plot, and thus we omit those plots.

Note that this prediction performance comparison does not consider any decision rejection options. In other words, both prediction algorithms use all predicted labels even though there exist many fewer confident predictions. In the following experiments, we investigate the effect of decision reject options on the prediction performance of models.

Experiment 2 (RQ3): Prediction with Rejection

To what extent do decision reject options influence the quality of predicting workers' next label? We conduct two experiments to examine the influence of varying the parameter δ representing a decision reject option. In this experiment, we also consider workers' actual correct/wrong observations as ground truth and measure the prediction performance over it. The score is individually computed for each worker, then averaged over all workers.

Figure 5 shows the effect of decision reject options on prediction performance along with varying original workers' accuracies. The original worker accuracy represents each worker's final running accuracy at the time point which each worker completed all the labeling task instances. The x-axis indicates the workers running accuracies; the y-axis shows F1-score of label prediction. With no rejection (when $\delta = 0$, the upper left figure), prediction performance is proportional to the increase of actual worker accuracy. It suggests that the prediction performance (F1 score) of both prediction models exceeds 0.8 for highly accurate workers ($>$0.7) while prediction is worse for relatively inaccurate workers ($\leq$0.7).

The other sub-figures in Figure 5 suggest that the increase of δ improves the performance of TS-based label predictions while no significant performance improvement is reported in RA-based label predictions. Decision rejection options get rid of label predictions with less confidence. Therefore, overall prediction performance improves with increased rejection parameter δ. In terms of the correlation between per-

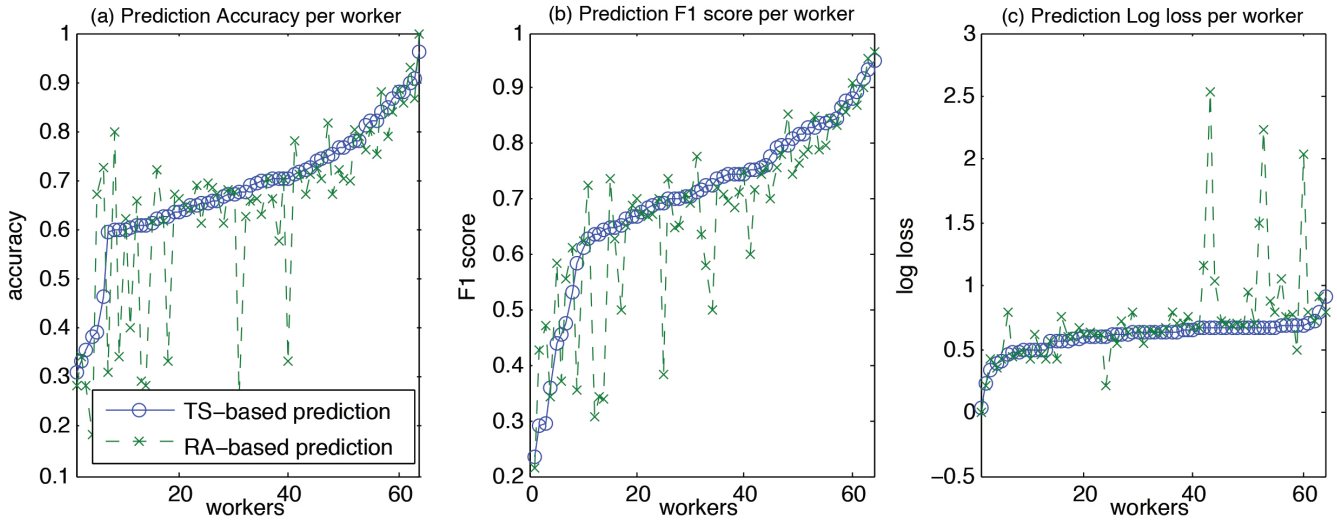

Figure 4: Predictions of workers' next labels are evaluated by Accuracy, F1, and LogLoss. Our time series(TS)-based prediction outperforms the running accuracy (RA)-based prediction for most workers across all three metrics.

formance predictions and actual workers' accuracies, decision reject options improve the prediction performance of less accurate workers (having accuracy between 0.5 and 0.7) more efficiently than the RA-based predictions.

RA-based prediction performance does not improve since the running accuracy does not reflect the dynamics of workers' correct/wrong patterns. In other words, the oscillation of running accuracy becomes smaller over time and thus the RA-based prediction is not able to capture the dynamics. On the contrary, TS-based prediction consider workers' correct/wrong pattern at each time point and therefore this model is able to predict a short-term label more accurately than the RA-based predictions. In addition, decision rejection options even lead to the further improvement of predicted label quality by the predicted time series model.

Finally, this experiment demonstrates that decision reject options significantly improve the quality of TS-based predicted labels by avoiding the risk of less confident predictions. The RA-based prediction does not achieve similar improvement as the TS-based prediction even though decision reject options are applied. We only report the prediction performance of label prediction vs. original workers' accuracies across varying decision reject options since the results measured by F1 and LogLoss also show similar patterns.

While further improvement of label predictions can be achieved by decision reject options, more conservative decisions not to predict naturally decrease the number of predictions made, as **Figure 6** shows. Without decision rejection, the number of predicted labels are 163 across all the workers. However, increasing δ decreases the number of predictions since there are many ambiguous $logit^{-1}(x_t)$. For instance, in case of $\delta = 0.05$, two prediction models reject their predictions if $0.45 < logit^{-1}(x_t) < 0.55$. Therefore, the increase of δ naturally decreases the number of prediction labels. However, the accuracies substantially increase by rejecting uncertain predictions. In terms of accuracy, the proposed TS-based prediction improves its performance from 0.65 to 0.82 while the RA-based prediction does not achieve any performance improvement. Besides, our proposed model shows similar quality improvement in terms of F1 score except with the highest setting of $\delta = 0.25$.

Experiment 3 (RQ4): Label Quality Improvement

Predicion method	TS	RA	Original Label
LogLoss	0.31**	0.42*	3.65
F1 score	0.79**	0.65*	0.62
Accuracy	0.79**	0.71*	0.59

Table 2: Label quality over ground truth. Decision rejection option was set $\delta = 0.2$. (**) indicates that TS-based prediction method outperforms the other two methods with high statistical significance ($p<0.05$). (*) indicates that SA-based method outperforms the quality of original labels with high statistical significance ($p<0.05$).

The previous two experiments showed our temporal model better predicting the next label of each worker than the baseline (RA-based prediction). Moreover, we demonstrated that decision reject options further improve the prediction performance by avoiding less confident predictions.

Next, we conduct an experiment on the quality of crowdsourced labels over ground truth generated by expert annotators. We measure three metrics (LogLoss, F1, and accuracy) in order to compare the quality of actual labels generated by two prediction methods (TS-based prediction vs. RA-based prediction) to original labels collected from workers without any task recommendation. For the experiment, we use *soft* (i.e., probabilistic) labels for LogLoss and *hard* (i.e., rounded, binary) labels for F1 and accuracy for *proper scoring rules* (Gneiting and Raftery 2007). In addition, we

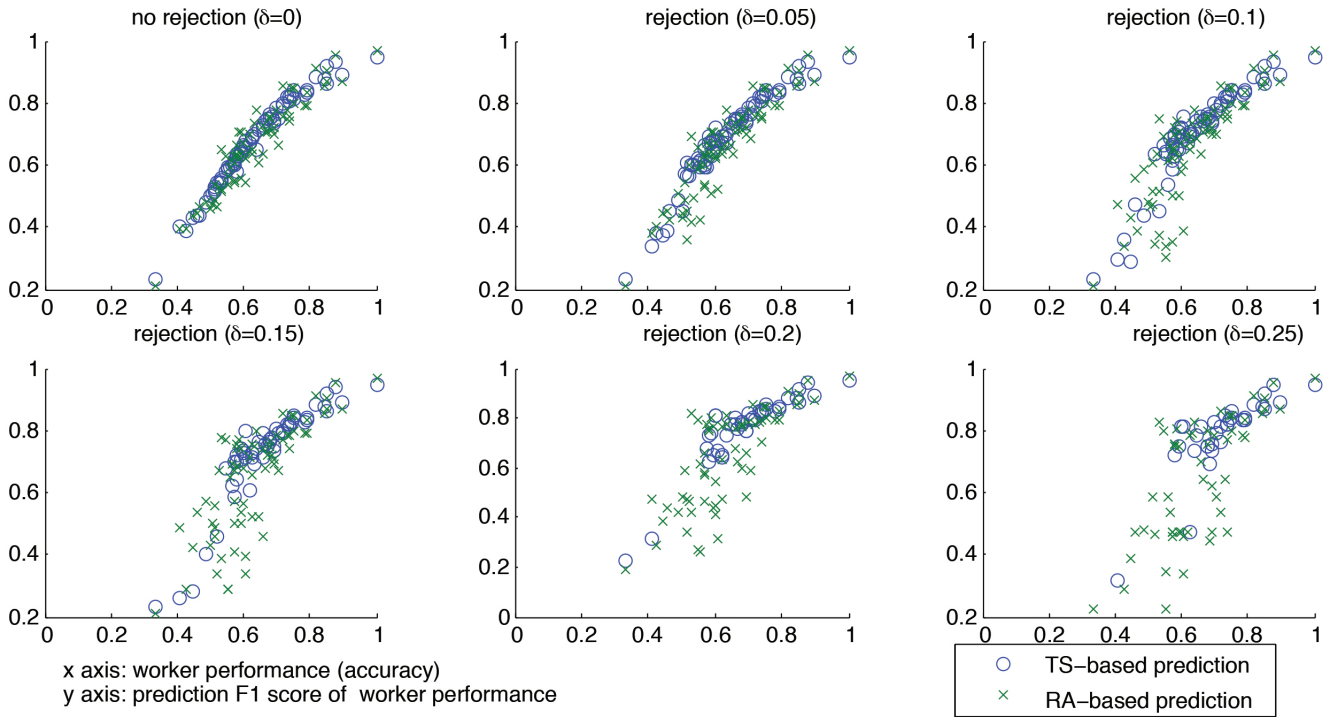

Figure 5: The benefit of predicting workers' next labels vs. simply relying on workers' *running accuracies* across different rejection strategies (varying $\delta \in [0, 0.25]$ by 0.05 step-size). As a tradeoff, by predicting only above a certain confidence level, increasing delta naturally increases prediction quality but decreases coverage (the number of predictions made). Increasing δ with TS-based prediction for medium-quality workers (near 0.5 accuracy) improves predictions substantially. RA-based prediction does not achieve similar improvement since it does not capture dynamicity of workers' labeling patterns.

conduct a paired-sample t-test in order to confirm a significance difference between prediction methods. A decision reject option ($\delta = 0.2$) is used for this experiment. Each score indicates average prediction score across all workers. For simplicity, we do not consider any aggregation methods.

Table 2 shows prediction scores of each method with respect to ground truth. Temporal modeling is seen to outperform the baseline (SA-based prediction) by 10-20% and significantly improve upon original labels by 20-30%. This suggests that label generation via our time-series prediction model leads to quality improvement of crowdsourced labels.

Conclusion and Future Work

Predicting the best workers can helpfully support successful task recommendation in crowdsourcing. While the existing studies make i.i.d. assumptions in terms of analyzing crowd workers' behavioral patterns for finding the best worker, we present a time-series prediction model in order to take account of the dynamics of workers' temporal behavioral patterns. Our experiments demonstrate that the proposed model not only predicts the actual workers' label more accurately but also improves the quality of crowdsourced labels over ground truth. This study presents a promising direction of time-series modeling to improve crowd work quality.

One direction to extend this study is to design an online time series prediction models. Although our parameter estimation algorithm is more efficient than a generic EM algorithm, it still requires several iterations to converge. To implement a real-time (e.g. millisecond time window) task recommendation system, however, our estimation algorithm can be a bottleneck. If we sacrifice the granularity of estimated parameters, however, the parameters can be estimated in a single pass of data. The basic idea is to restrict our parameters to be from finite sets e.g. $c \in \{-1, -0.9, \ldots, 0.9, 1\}$. Then, our particle filter could be applied with a finite set of parameter combinations in parallel, and we pick the parameter set that provides the maximum pseudo-likelihood of the time series. This large-scale, real-time implementation is worth investigating in future work to empirically verify the trade-off between parameter resolution and worker recommendation performance.

References

Bartlett, P. L., and Wegkamp, M. H. 2008. Classification with a reject option using a hinge loss. *J. Mach. Learn. Res.* 9:1823–1840.

Bernstein, M. S.; Karger, D. R.; Miller, R. C.; and Brandt, J. 2012. Analytic methods for optimizing realtime crowdsourcing. In *Collective Intelligence*.

Box, G.; Jenkins, G. M.; and Reinsel, G. C. 1994. *Time Series Analysis: Forecasting and Control*. Prentice-Hall, third edition.

Buckley, C.; Lease, M.; and Smucker, M. D. 2010. Overview of the TREC 2010 Relevance Feedback Track

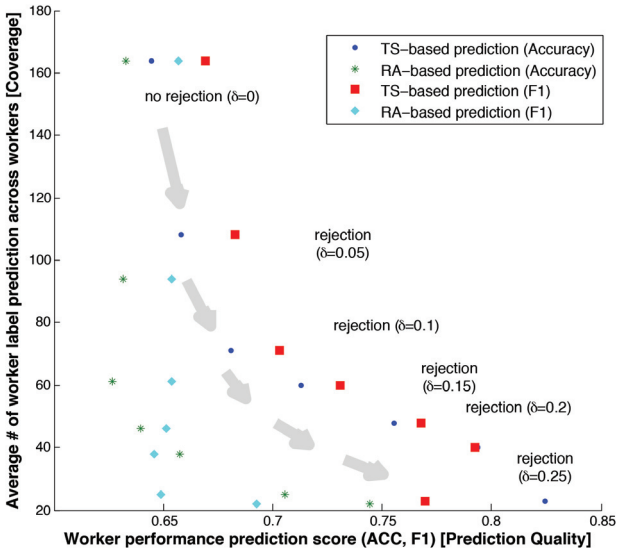

Figure 6: Prediction Performance (accuracy and F1) of workers' next label and its coverage across varying decision rejection options (δ=[0 0.25] by 0.05). The increase of δ improves the quality of the TS-based prediction while sacrificing the average number of predictions (coverage). The coverage of RA-based predictions also decreases but its prediction quality is not improved substantially compared to the TS-based prediction. The TS-based prediction outperforms the RA-based prediction in terms of quality and coverage. Furthermore, decision reject options improve the quality of TS-based prediction by trading-off prediction coverage.

(Notebook). In *The Nineteenth Text Retrieval Conference (TREC) Notebook*.

Burg, J. P. 1967. Maximum entropy spectral analysis. In *Proc. 37th Meeting of the Society of Exploration Geophysicists*.

Canova, F., and Cicarelli, M. 2013. Panel vector autoregressive models: A survey. *European Central Bank: Working Paper Series*.

Carterette, B., and Soboroff, I. 2010. 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, 539–546. New York, NY, USA: ACM.

Cosley, D.; Frankowski, D.; Terveen, L.; and Riedl, J. 2007. Suggestbot: using intelligent task routing to help people find work in wikipedia. In *12th ACM IUI Conference*, 32–41.

Dai, P.; Mausam; and Weld, D. S. 2010. Decision-theoretic control of crowd-sourced workflows. In *Proc. AAAI*.

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

Engle, R. F. 1982. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica* 50(4):987–1007.

Gneiting, T., and Raftery, A. E. 2007. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association* 102:359–378.

Grady, C., and Lease, M. 2010. Crowdsourcing document relevance assessment with mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, 172–179. Stroudsburg, PA, USA: Association for Computational Linguistics.

Jacobs, P. A., and Lewis, P. A. W. 1983. Stationary discrete autoregressive-moving average time series generated by mixtures. *Journal of Time Series Analysis* 4(1):19–36.

Juang, B. H., and Rabiner, L. R. 1991. Hidden Markov Models for speech recognition. *Technometrics* 33(3):251–272.

Kamar, E.; Hacker, S.; and Horovitz, E. 2012. Combining human and machine intelligence in large-scale crowdsourcing. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 467–474.

Kaufmann, H. 1987. Regression models for nonstationary categorical time series: asymptotic estimation theory. *The Annals of Statistics* 15(1):79–98.

Kittur, A.; Nickerson, J.; Bernstein, M. S.; Gerber, E.; Shaw, A.; Zimmerman, J.; Lease, M.; and Horton, J. J. 2013. The future of crowd work. In *In Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW)*, 1301–1318.

Law, E.; Bennett, P.; and Horvitz, E. 2011. The effects of choice in routing relevance judgments. In *Proceedings of the 34th ACM SIGIR*, 7–8.

Li, H.; Zhao, B.; ; and Fuxman, A. 2014. The Wisdom of Minority: Discovering and Targeting the Right Group of Workers for Crowdsourcing. In *Proceedings of the 23rd WWW conference*.

Litterman, R. B. 1984. Specifying vector autoregressions for macroeconomic forecasting. *Federal Reserve Bank of Minneapolis Staff report* 1(92).

Nadeem, M. S. A.; Zucker, J.-D.; and Hanczar, B. 2010. Accuracy-rejection curves (arcs) for comparing classification methods with a reject option. In *Machine Learning in System Biology, Journal of Machine Learning*, volume 8 of *JMLR Proceedings*, 65–81.

Park, Y.; Carvalho, C.; and Ghosh, J. 2014. Lamore: A stable, scalable approach to latent vector autoregressive modeling of categorical time series. In *17th International conference AISTAT*.

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

Pillai, I.; Fumera, G.; and Roli, F. 2013. Multi-label classification with a reject option. *Pattern Recognition* 46(8):2256 – 2266.

Raftery, A. E. 1985. A model for high-order markov chains. *Journal of the Royal Statistical Society. Series B (Methodological)* 47(3):528–539.

Shahaf, D., and Horvitz, E. 2010. Generalized task markets for human and machine computation. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, 986–993.

Viterbi, A. J. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Information Theory* 13(2):260–269.

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

Yuen, M.; King, I.; and Leung, K.-S. 2012. Task recommendation in crowdsourcing systems. In *Proceedings of the First International Workshop on Crowdsourcing and Data Mining*, 22–26.

Zeger, S. L.; Liang, K.-Y.; and Albert, P. S. 1988. Models for longituidinal data: A generalized estimating equation approach. *Biometrics* 44:1049–1060.

Zhen, X., and Basawa, I. V. 2009. Observation-driven generalized state space models for categorical time series. *Statistics and Probability Letters* 79:2462–2468.

Zucchini, W., and MacDonald, I. L. 2009. *Hidden Markov Models for Time Series: An Introduction Using R*. Chapman and Hall/CRC.