Proceedings, The Fourth AAAI Conference on
Human Computation and Crowdsourcing

Predicting Crowd Work Quality under Monetary Interventions

Ming Yin
Harvard University
mingyin@fas.harvard.edu

Yiling Chen
Harvard University
yiling@seas.harvard.edu

Abstract

Work quality in crowdsourcing task sessions can change over time due to both internal factors, such as learning and boredom, and external factors like the provision of monetary interventions. Prior studies on crowd work quality have focused on characterizing the temporal behavior pattern as a result of the internal factors. In this paper, we propose to explicitly take the impact of external factors into consideration for modeling crowd work quality. We present a series of seven models from three categories (supervised learning models, autoregressive models and Markov models) and conduct an empirical comparison on how well these models can predict crowd work quality under monetary interventions on three datasets that are collected from Amazon Mechanical Turk. Our results show that all these models outperform the baseline models that don't consider the impact of monetary interventions. Our empirical comparison further identifies the random forests model as an excellent model to use in practice as it consistently provides accurate predictions with high confidence across different datasets, and it also demonstrates robustness against limited training data and limited access to the ground truth.

Introduction

In recent years, crowdsourcing brings up significant benefits in a wide range of domains by eliciting contribution from human workers in an affordable, scalable and on-demand manner (Greengard 2011). To further improve the productivity of crowdsourcing, one direction that researchers and practitioners are actively working on is to understand the actual worker behavior within crowdsourcing systems, for example, to model worker performance in a crowdsourcing task session. Along this direction, while early work often focuses on estimating a worker's inherent capability level (sometimes referred to as the error rate) which is *independent* of the working environment, does *not* change over time and determines the worker's performance in the tasks (Whitehill et al. 2009; Karger, Oh, and Shah 2011; Raykar and Yu 2012), recent work suggests that worker performance can be better modeled when taking its time-variance (e.g. improvement or degradation over time) into consideration (Donmez, Carbonell, and Schneider 2010; Jung, Park, and Lease 2014; Bragg and Weld 2016).

Copyright © 2016, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

The time-variance of crowd work quality discussed in these studies often describes the *organic evolvement* of worker performance, perhaps due to the learning effect or boredom. In reality, however, worker performance can also be influenced by some external factors presented in the working contexts, such as certain *interventions* embedded in the task session. Examples of interventions in crowdsourcing systems include the placement of extra monetary rewards (Harris 2011; Ho et al. 2015; Yin, Chen, and Sun 2013), the provision of performance feedback (Dow et al. 2012) and the insertion of short breaks between subsequent tasks (Dai et al. 2015). Then, it is a natural question to ask how should we model the fluctuation of worker performance in a crowdsourcing task session, given both the organic evolvement and the impact of external interventions.

As granting extra bonuses to workers is a common approach that requesters use to encourage high-quality work, in this study, we choose to focus on characterizing the impact of *monetary interventions* on worker performance. The goal of this paper is thus to model how crowd work quality changes in a task session given monetary intervention on selected tasks. Properly addressing this problem is, in fact, quite meaningful for the requesters — As shown in Yin and Chen (2015), a good model of worker performance under monetary intervention will enable a requester to dynamically control the provision of monetary interventions such that compared to randomly placing the bonuses, the requester can obtain significantly more high-quality work with less cost.

In this paper, we take a *prediction* perspective for the crowd work quality modeling problem. That is, given a specific type of tasks at hand, we assume that the requester has already recruited several workers to work on them sequentially in task sessions. The requester records whether monetary intervention is provided as well as the work quality on each task for each worker. In this way, the requester essentially obtains a *training dataset* of worker behavior from which he can reason about how crowd work quality changes with the provision of monetary interventions. Then, for a new worker who starts to work on a task session, after monitoring the worker's performance for a short period of time, the requester aims at predicting the worker's performance in the current task, given whether monetary intervention is provided on this task as well as the history of monetary in-

tervention provisions and work quality for all previous tasks that the worker has already completed in the session. As work quality is usually measured with discrete levels (e.g. high-quality or low-quality) and is likely to be influenced by the provision of monetary interventions, this problem is essentially a *categorical time series prediction with exogenous inputs*.

We address this prediction problem with 7 models from 3 different categories, and present an empirical comparison on how well these models perform. Specifically, we first treat our prediction as a classification problem and adopt three *supervised learning models* (random forests, support vector machine and artificial neural network). Furthermore, we propose two time-series models (DARX and LARX) that are extended from existing *autoregressive models* to incorporate the exogenous inputs. Finally, by assuming that the change of work quality (or the change of some latent variable related to work quality) is governed by a Markov process, we consider two variants of the *Markov models* (controlled Markov chain and input-output hidden Markov model) for our prediction. The performance of each model is examined on three datasets that are collected with *real* crowd workers from Amazon Mechanical Turk (MTurk) for different types of tasks, including solving word puzzles, classifying images, and finding typos in the text.

In addition, requesters often face some practical constraints when predicting crowd work quality: (1) *the "cold start" problem*: requesters have very limited training data to start with, hence their knowledge on how workers react to monetary interventions is quite limited at the beginning; (2) *the lack of ground truth*: requesters often get access to the ground truth for only a certain number of tasks, hence they can only evaluate a worker's performance on *some* tasks in the past when making prediction on her work quality in the current task. Therefore, to better understand the robustness of the models when facing realistic constraints, we conduct further experiments to investigate the performance of different prediction models when the requester has limited training data or limited ground truth.

Our results first confirm the necessity to explicitly take the effects of monetary interventions into consideration when modeling crowd work quality. In particular, compared to the two baseline models (running accuracy and latent autoregressive) which only characterize the organic evolvement of worker performance, the seven proposed models can almost always make more accurate predictions with higher confidence for the crowd work quality under monetary interventions. Furthermore, our empirical comparison among different models suggests that the random forests model is an excellent model to use for predicting crowd work quality under monetary interventions in practise. On the one hand, the random forests model presents a *consistently* high prediction performance across various datasets; on the other hand, unlike some of the other proposed models, the random forests model demonstrates *robust* prediction performance, despite of the relatively small training datasets or the limited access to ground truth.

Related Work

Most existing work on modeling temporal crowd work quality focuses on capturing the organic evolvement of worker performance, and various time-series models have been adopted in these studies. For example, Donmez, Carbonell, and Schneider (2010) proposed a Bayesian time series model, assuming that the latent variable dynamics that governs the change of work quality over time has an uniform offset and correlation, that is $x_t = x_{t-1} + \epsilon_t$. Jung, Park, and Lease (2014) relaxed this constraint and came up with a generalized model (LAR) with $x_t = c + \phi x_{t-1} + \epsilon_t$. More recently, Jung and Lease (2015a) designed a generalized time-varying assessor model (GAM) that is a logistic regression predictor with features extracted from both generative time-series models (e.g. the estimated ϕ and c from the LAR model) and worker's behavioral evidence, and they showed that the prediction accuracy on crowd work quality can be significantly improved with this model. Meanwhile, Bragg and Weld (2016) took a different approach and used a parametric hidden Markov model to explicitly model the performance degradation over time.

Although many studies have explored the impact of various external interventions on the performance of crowd workers (Mason and Watts 2010; Shaw, Horton, and Chen 2011; Huang and Fu 2013), few work has been done on modeling temporal crowd work quality given the presence of external (monetary) interventions. The closest prior work we are aware of, by Yin and Chen (2015), used a first-order input-output hidden Markov model (IOHMM) to characterize how work quality is influenced by bonus over time, although the focus of that paper was on using the learned IOHMM to dynamically control the placement of bonus, hence the authors didn't report on the prediction performance of their model. In this paper, we enumerate 7 models from 3 categories and conduct an empirical comparison of their performance in predicting crowd work quality under monetary interventions. While we adopt some existing models from the literature (e.g. supervised learning models, Markov model variants), we also propose a few new models. In particular, not many time-series models are known for the categorical prediction problems with exogenous input sequence as ours — models like discrete autoregressive (DAR) and latent autoregressive (LAR) deal with categorical time series predictions *without* exogenous inputs (Jacobs and Lewis 1983; Jung, Park, and Lease 2014), while models like autoregressive with exogenous inputs (ARX) deal with predictions with exogenous inputs for *continuous* variables (Ljung 1998). Hence, we propose two variants of autoregressive models, DARX and LARX, for our prediction problem, which are extended from the existing models. We further study the performance of these prediction models in more realistic scenarios, such as when the requester has limited training dataset or limited ground truth. Similar analyses have been conducted previously in different contexts, for the prediction of disengagement (Mao, Kamar, and Horvitz 2013) or predicting temporal work quality without external interventions (Jung and Lease 2015b).

Finally, notice that while we focus on the prediction of work quality under monetary interventions in this study, the

models in our paper can be easily generalized to predict other crowd worker behavior of interests (e.g. engagement) given other types of interventions, such as the provision of feedback (Dow et al. 2012), the switch of workflows (Lin, Mausam, and Weld 2012) and the deliver of communication messages (Segal et al. 2016).

Prediction Models

Our prediction problem can be formally defined as the following: The requester has collected a training dataset $\mathbb{D}_{train}$ of N workers. Each worker in the training dataset completes a sequence of T tasks. For each worker $i(1 \leq i \leq N)$, the requester keeps a record of the sequence of monetary interventions provided to the worker $\boldsymbol{a_i} = (a_i^1, a_i^2, \cdots, a_i^T)$ as well as the sequence of observed work quality $\boldsymbol{y_i} = (y_i^1, y_i^2, \cdots, y_i^T)$. For simplicity, we consider binary levels of monetary interventions and work quality in this paper. That is, $a_i^t \in \{0, 1\}(1 \leq t \leq T)$ indicates whether a monetary intervention is provided on task t to worker i, with value 1 (or 0) representing a positive (or negative) answer, and $y_i^t \in \{0, 1\}$ refers to the work quality of worker i on task t, with value 1 (or 0) representing high-quality (or low-quality) work. The requester is interested in modeling crowd work quality under monetary interventions through the training dataset and making predictions for a future worker — given the sequence of monetary interventions $\boldsymbol{a} = (a^1, a^2, \cdots, a^{l-1})$ provided to this worker so far as well as the observed work quality $\boldsymbol{y} = (y^1, y^2, \cdots, y^{l-1})$, what's the worker's performance y^l in the current task (i.e. the l-th task) when monetary intervention level a^l is provided?

Supervised Learning Models

We first treat our prediction as a supervised learning problem. Take worker i's performance in task t for an example, y_i^t is naturally the *label* for this training instance. We further extract a *feature set* for this instance by focusing on a history window of size L. In particular, the feature set $\boldsymbol{x_i^t}$ includes:

- *current intervention level*: a_i^t, whether a monetary intervention is provided in the current task;
- *average intervention level*: $\frac{1}{t-1}\sum_{j=1}^{t-1} a_i^j$, the percentage of tasks with monetary interventions among all previous tasks;
- *average performance*: $\frac{1}{t-1}\sum_{j=1}^{t-1} y_i^j$, the percentage of high-quality work in all previous tasks;
- *historical intervention levels*: $a_i^h(t - L \leq h \leq t - 1)$, whether monetary intervention is provided in each of the previous L tasks;
- *historical performance*: $y_i^h(t-L \leq h \leq t-1)$, the work quality in each of the previous L tasks;
- *historical intervention changes*: $a_i^{h_2} - a_i^{h_1}(t-L \leq h_1 < h_2 \leq t - 1)$, the differences in monetary interventions for any two of the previous L tasks; and
- *historical performance changes*: $y_i^{h_2} - y_i^{h_1}(t-L \leq h_1 < h_2 \leq t - 1)$, the differences in work quality for any two of the previous L tasks.

A transformed training dataset is created through extracting the feature-label pair $(\boldsymbol{x_i^t}, y_i^t)$ for all workers and all tasks in the original training dataset. A supervised learning model then simply constructs a function $y_i^t = f(\boldsymbol{x_i^t})$ that maps the features to the label. We consider three such model in this paper:

Model 1: Random Forests (RF) Random forests (Ho 1998) is a popular ensemble learning technique for classification and regression. Briefly speaking, many decision trees are grown in the random forests. Each tree is constructed by fitting a decision tree to a *random* subset of the training data, and a *random* subset of features are considered for each split within the tree. The prediction for a testing sample is made by classifying it using each decision tree in the forest in turn and then taking the majority vote among all trees.

Model 2: Support Vector Machine (SVM) The general idea of support vector machine (Cortes and Vapnik 1995) is to map training data points from the original finite-dimensional space to a higher-dimensional space, and search for a *hyperplane* to separate data points from different classes such that the distance between the closest two data points of different classes is maximized. *Kernel functions* are often used to construct non-linear SVM classifiers (Boser, Guyon, and Vapnik 1992). To make a prediction for a testing sample, we simply map it to the same higher-dimensional space and assign a label to it according to on which side of the hyperplane it falls.

Model 3: Artificial Neural Network (NN) Inspired by the biological neural networks, artificial neural networks are a family of machine learning models that can approximate any function between features and labels (Hornik, Stinchcombe, and White 1989). While various network structure can be designed based on the understanding of the specific prediction problem, in this study, we focus on a fully connected multi-layer neural network — In this network, there is a layer of *input* neurons, a layer of the single *output* neuron and one or more layers of *hidden* neurons where each neuron in one hidden layer is connected to *all* neurons in the previous (input or hidden) layer as well as *all* neurons in the next (hidden or output) layer. Specifically for our problem, each element in the feature set $\boldsymbol{x_i^t}$ activates an input neuron and the single output neuron produces the label y_i^t. A neuron in a hidden layer takes the weighted sum of output values from the previous layer as the input, and outputs a value after transforming the input with an *activation function*. The weights between any two neurons in the network are estimated through the training data. The prediction of a testing sample can be completed by feeding the input neurons with its features, activating hidden neurons in turn and determining the label until the output neuron is activated.

Autoregressive Models

Next, we introduce two variants of the autoregressive models in time series analysis to address our prediction problem.

Model 4: Discrete Autoregressive Model with Exogenous Inputs (DARX) We extend the Discrete Autoregressive (DAR) model (Jacobs and Lewis 1983) to incorporate the exogenous inputs. Formally, a DARX model of order p is defined as follows:

$$y_i^t = I_t y_i^{t-D_t} + (1 - I_t)e_t \tag{1}$$

where e_t is a binary variable with $Pr(e_t = 1|a_i^t) = \beta_{a_i^t}$, I_t is a binary variable with $Pr(I_t = 1|a_i^t) = \lambda_{a_i^t}$, D_t randomly takes a value from the set $\{1, 2, \cdots, p\}$ with $Pr(D_t = d|a_i^t) = \alpha_{a_i^t}^d$, and $\sum_{d=1}^p \alpha_{a_i^t}^d = 1$ for $a_i^t \in \{0, 1\}$. Importantly, notice that in the DARX(p) model, the probability distributions for random variables e_t, I_t and D_t are all *conditioned* on the exogenous input a_i^t. This is different from the DAR model where exogenous inputs are not included as a part. As a concrete example, consider when monetary intervention is provided to worker i on task t, that is, $a_i^t = 1$. Then, the DARX(p) model states that, the value of y_i^t (i.e. whether worker i will submit high-quality work on task t) is related to the previously observed work quality with probability λ_1 (i.e. when $I_t = 1$) and not related with probability $1 - \lambda_1$ (i.e. when $I_t = 0$). When $I_t = 0$, y_i^t is determined by an independent binary variable e_t, which is equal to 1 with probability β_1. On the other hand, when $I_t = 1$, y_i^t equals to the observation $d(1 \leq d \leq p)$ steps ago, that is, y_i^{t-d}, with probability α_1^d.

The DARX(p) model has $2p+4$ parameters to estimate in total: λ_a, α_a^d and β_a, with $a \in \{0, 1\}$ and $d \in \{1, 2, \cdots, p\}$. Given the training dataset, we can search for a set of parameters that best characterizes worker's reaction to monetary interventions as a *population*. To make a prediction for a testing worker with these population-level parameters on her l-th task, we simply draw random variables e_l, I_l and D_l according to the estimated parameters and decide the label of the testing sample with Equation 1.

On the other hand, parameters of a DARX(p) model can also be estimated in an online fashion for the *individual* worker that we are currently predicting on. This enables us to make more personalized predictions — We may initialize the model with the population-level parameters, that is, $\lambda_a^1 = \lambda_a$, $\alpha_a^{d,1} = \alpha_a^d$ and $\beta_a^1 = \beta_a$, and we can update these parameters over time as we keep observing the testing worker completes more tasks in the session and obtaining the individual-level model estimates. One way to update the model parameters is to take a weighted average of the old parameters and the newly estimated individual-level parameters at each time step. For instance, suppose the testing worker has completed a sequence of $l-1$ tasks and the observed sequences of $\boldsymbol{a}$ and $\boldsymbol{y}$ lead to an individual-level model with parameters $\lambda_a^{'}$, $\alpha_a^{d^{'}}$ and $\beta_a^{'}$. We propose to update the model parameters as the following:

$$\lambda_a^l = (1 - \gamma)\lambda_a^{l-1} + \gamma\lambda_a^{'} \tag{2}$$

$$\alpha_a^{d,l} = \frac{(1 - \gamma)\lambda_a^{l-1}\alpha_a^{d,l-1} + \gamma\lambda_a^{'}\alpha_a^{d^{'}}}{\lambda_a^l} \tag{3}$$

$$\beta_a^l = \frac{(1 - \gamma)(1 - \lambda_a^{l-1})\beta_a^{l-1} + \gamma(1 - \lambda_a^{'})\beta_a^{'}}{1 - \lambda_a^l} \tag{4}$$

The prediction for the l-th task is made based on $\lambda_a^l, \alpha_a^{d,l}$ and β_a^l, and a new set of individual-level parameters will be estimated after we observe the actual work quality y^l in the l-th task. Notice that $\gamma(0 \leq \gamma \leq 1)$ represents the *learning rate* for parameter updating: When $\gamma = 0$, the prediction is always made with population-level parameters, and when $\gamma = 1$, the prediction is made with individual-level parameters exclusively.

Model 5: Latent Autoregressive Model with Exogenous Inputs (LARX) The second autoregressive model variant is extended from the Latent Autoregressive Model (LAR) (Jung, Park, and Lease 2014). Specifically, the LAR model is defined as follows:

$$z_i^t = c + \phi z_i^{t-1} + \epsilon_i^t \tag{5}$$

$$Pr(y_i^t = 1) = \frac{1}{1 + e^{-z_i^t}} \tag{6}$$

where $\epsilon_i^t \sim N(0, \sigma^2)$ is a random noise, z_i^t is a latent variable that governs the worker's performance and the observed work quality y_i^t is determined stochastically by z_i^t through the logistic function. To take the impact of monetary interventions on work quality into consideration, we propose a generalized LARX model, with an autoregressive order of p and an exogenous input order of q, by replacing Equation 5 with the following formula:

$$z_i^t = c + \sum_{j=1}^{p} \phi_j z_i^{t-j} + \sum_{j=0}^{q-1} \theta_j a_i^{t-j} + \epsilon_i^t \tag{7}$$

Equation 7 is essentially an autoregressive model with exogenous inputs (ARX) (Ljung 1998). Different from the LAR model, the LARX model assumes that the latent variable z_i^t depends linearly on both its previous values and the exogenous inputs. Given the training dataset, a population-level LARX model can be learned through expectation-maximization algorithms with particle filters (Park, Carvalho, and Ghosh 2014). While the population-level model can be used for prediction, similar to the DARX model, we can also make more personalized predictions by updating the LARX model parameters over time (e.g. $\phi_j^l = (1-\gamma)\phi_j^{l-1} + \gamma\phi_j^{'}$) to characterize both the population-level behavior and the individual-level behavior.

Markov Models

Finally, we present two Markov models for predicting crowd work quality under monetary interventions.

Model 6: Controlled Markov Chain (CMC) Controlled Markov chain includes exogenous inputs (often referred to as "actions") into a Markov chain, and with further addition of reward functions, a CMC will be transformed into a Markov decision process (MDP). A CMC of order p defines that state transition depends only on the recent p states and the current input, that is, $P_a(S_p, \cdots, S_1, S_0) = Pr(s_t = S_0|s_{t-1} = S_1, \cdots, s_{t-p} = S_p, a_t = a) = Pr(s_t = S_0|s_{t-1} = S_1, \cdots, s_1 = S_{t-1}, a_t = a)$. For our purpose, we take the observed work quality in each task as the "state". Thus, the state transition probabilities essentially represent

the distribution of the work quality y_i^t in task t, given the monetary intervention level a_i^t in task t and the observed work quality sequence $(y_i^{t-p}, y_i^{t-p+1}, \cdots, y_i^{t-1})$ in the past p tasks. A maximum-likelihood estimate of these transition probability parameters can be obtained given the training dataset. For the testing worker, we predict that $Pr(y^l = 1) = P_{a^l}(y^{l-p}, y^{l-p+1}, \cdots, y^{l-1}, 1)$.

Model 7: Input-Output Hidden Markov Model (IOHMM) Input-output hidden Markov model (Bengio and Frasconi 1995) is a variant of the hidden Markov model for learning the mapping between input and output sequences. An IOHMM of order p is defined as follows:

- *inputs*: a_i^t, whether a monetary intervention is provided in task t;
- *outputs*: y_i^t, the work quality in task t;
- *hidden states*: $z_i^t \in \{1, 2, \cdots, K\}$, the worker's latent state in task t, where K is the total number of hidden states;
- *transition probability*: $P_{tr}(z_i^t | z_i^{t-1}, \cdots, z_i^{t-p}, a_i^t)$, the probability of transiting to state z_i^t in task t given the current input a_i^t and state sequence $(z_i^{t-p}, z_i^{t-p+1}, \cdots, z_i^{t-1})$ in the previous p tasks; and
- *emission probability*: $P_e(y_i^t | z_i^t, \cdots, z_i^{t-p+1}, a_i^t)$, the probability of submitting work of quality y_i^t in task t given the current input a_i^t and the state sequence $(z_i^{t-p+1}, \cdots, z_i^{t-1}, z_i^t)$ in the recent p tasks.

An IOHMM can be estimated using the Baum-Welch expectation-maximization algorithm (Bengio and Frasconi 1996). To make predictions for the testing worker, we maintain and update a state belief $\boldsymbol{b}_1 (1 \leq 1 \leq l)$ at each step, which is the probability distribution for the worker to stay in different combinations of states in the p tasks before task 1. The value of y^l is then computed with $\boldsymbol{b_l}$ and a^l. For example, when $p = 1$, we have $\boldsymbol{b_l} = (b_l(1), b_l(2), \cdots, b_l(K))$ where $b_l(k) (1 \leq k \leq K)$ is the estimated probability for the worker to stay in hidden state k in task $l-1$. Then, we predict that $Pr(y^l = 1) = \sum_{k=1}^K b_l(k) (\sum_{j=1}^K P_{tr}(j|k, a^l) P_e(1|j, a^l))$, and after we observe y^l, the state belief is updated according to that $b_{l+1}(j) \propto \sum_{k=1}^K b_l(k) P_{tr}(j|k, a^l) P_e(y^l|j, a^l)$. A first-order IOHMM is used in Yin and Chen (2015) to characterize the impact of bonus provisions on crowdsourcing work quality.

Datasets

We examine the performance of different prediction models on 3 datasets that are collected from real crowd workers in Amazon Mechanical Turk (MTurk):

- PUZZLE: consists of 300 workers each completing a sequence of 9 word puzzle tasks in one HIT. In each task, the worker is shown a "target" word as well as a 12×12 board filled with capital letters. The worker is asked to find the appearances of the target word on the board as many times as possible. The base payment for the HIT is 45 cents. The requester provides extra performance-contingent bonus on 37% of the tasks. When a worker submits a high-quality answer in a bonus task by pointing out more than 80% of all appearances of the target word, she can earn an extra bonus of 5 cents. This dataset is supplied by Yin and Chen (2015).
- CLASSIFY: consists of 220 workers each completing a sequence of 10 butterfly classification tasks in one HIT. In each task, the worker sees 5 pictures of butterflies and is asked to classify each picture into three categories of interests: black swallowtail, monarch and machaon. The base payment for the HIT is 50 cents. 29% of the tasks come with extra bonus. When the worker submits a high-quality answer in a bonus task by correctly classifying all 5 pictures in that task, she can earn an extra bonus of 5 cents. This dataset is collected by us and the set of butterfly pictures used in the tasks is taken from Lazebnik, Schmid, and Ponce (2004).
- TYPO: consists of 80 workers each completing a sequence of 10 typo-finding tasks in one HIT. In each task, there is a short paragraph of about 200 words. The worker is asked to proofread it and find out as many typos as possible. The base payment for the HIT is 1 dollar. In 49% of all the tasks, there are extra performance-contingent bonuses. If the worker submits a high-quality answer in a bonus task by finding out more than 75% of all the typos, she will earn a bonus of 10 cents. This dataset is collected by us and a similar task has been used in the previous study (Ho et al. 2015).

Results

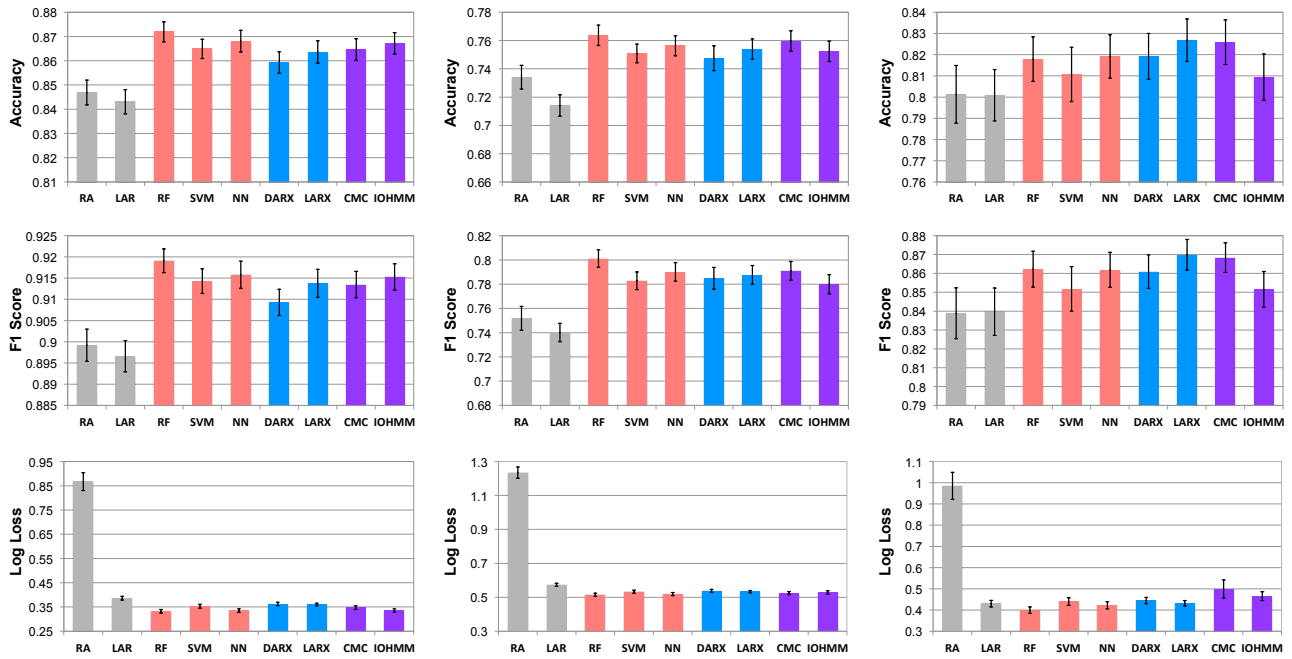
In this section, we report our empirical comparison results on the performance of different models in predicting the crowd work quality under monetary interventions.

Experimental Settings

Given a dataset, we first randomly take 80% of the workers in it and collect their data as the training dataset, while the data for the rest 20% of the workers is used as the testing dataset. For a particular model type (e.g. random forests), we fit a model of that type using the training dataset, and then use the estimated model to make predictions for each worker in the testing dataset. Since predicting the work quality in one task often rely on information about previous tasks, we start making predictions from the fourth task of each sequence. This process is repeated for 20 times, and the average performance of each prediction model across the 20 random splits is then reported.

Baselines For comparison, in our experiment, we include two baseline models that consider the organic evolvement of worker performance only:

- *running accuracy (RA)*: $Pr(y^l = 1) = \frac{1}{l-1} \sum_{j=1}^{l-1} y^j$, that is, the prediction on the l-th task is made according to the percentage of high-quality work observed in the previous $l-1$ tasks; and



(a) PUZZLE (b) CLASSIFY (c) TYPO

Figure 1: Performance comparisons for all prediction models on the three datasets (Top row: accuracy; middle row: F_1 score; bottom row: log loss). Means and standard errors of the mean are reported given 20 random splits of training and testing data.

- *latent autoregressive (LAR)*: the time-series model proposed by Jung, Park, and Lease (2014)[1].

Metrics We use 3 metrics to evaluate the performance of a prediction model:

- *accuracy*: the percentage of tasks in the testing dataset for which the prediction is correct;
- F_1 *score*: the harmonic mean of precision and recall, i.e. $\mathrm{F}_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision+recall}}$; and
- *log loss*: $-\frac{1}{N_{test}} \sum_{j=1}^{N_{test}} y_j \log(p_j) + (1-y_j) \log(1-p_j)$, where N_{test} is the total number of predictions made for the testing dataset, y_j is the true label of the j-th sample, and $p_j = Pr(y_j = 1)$ is the predicted probability of high-quality work for the j-th sample.

Intuitively, the higher the accuracy, the better the model. As some of our datasets are imbalanced[2], we provide the F_1 score for further reference. For prediction models that generate probabilistic labels (e.g. RA, LAR, DARX, LARX, CMC and IOHMM), in order to calculate accuracy and F_1 score, we assign a binary label to a testing sample according to a predefined threshold of 0.5, that is, $\hat{y}_j = 1$ when $p_j > 0.5$. Log-loss describes not only whether the prediction is accurate but also whether the prediction is confident, with a smaller value indicating a better model.

Model Selection Model selection is conducted through cross validation. Specifically, we partition the training dataset into 5 folds, pick each of the five folds to test while using the rest four folds to train models. The model setting with the highest average performance across the five folds (according to log loss) is then selected and a final model is trained with the whole training dataset using this setting.

We fix the size of history window $L = 3$ for all supervised learning models. For RF, we fix the number of trees to be 1,000 and tune on the minimum number of samples on a leaf; for SVM, we tune on the choice of kernel function (e.g. linear, polynomial, radial basis function, sigmoid); and for NN, we tune on the choice of activation function (e.g. logistic sigmoid, hyperbolic tan, rectified linear), the number of hidden layers (1 or 2) and the number of neurons in each hidden layer. For autoregressive models, we experiment with different learning rates $\gamma \in \{0, 0.01, 0.05, 0.1, 1\}$ for both DARX and LARX. While we also tune on the autoregressive order ($p \in \{1, 2, 3\}$) for DARX, to have a direct comparison between LAR and LARX, we set $p, q = 1$ for LARX. Finally, for the Markov models, we experiment with 3 types of CMC with $p \in \{1, 2, 3\}$ and 4 types of IOHMM: first-order IOHMMs with different number of hidden states $K \in \{2, 3, 4\}$, and a second-order IOHMM with $K = 2$.

A Comparison on Prediction Performance

Figure 1 compares the prediction performance of all 9 models (2 baseline models and 7 proposed models) on the three datasets. We first observe that the 7 proposed models al-

[1]Although GAM is proposed as an improvement of LAR in Jung and Lease (2015a), we can not use GAM as a baseline because GAM is tailored to their specific dataset.

[2]The percentages of tasks with high-quality work in the PUZZLE, CLASSIFY and TYPO datasets are 76.8%, 55.5% and 63.4%, respectively.

Table 1: Performance comparison between random forests (RF) and other prediction models. The differences in mean values for each metric are reported. The statistical significance of paired t-test is marked as a superscript, with †, *, **, and *** representing significance levels of 0.1, 0.05, 0.01, and 0.001 respectively.

Metric	Dataset	RA	LAR	SVM	NN	DARX	LARX	CMC	IOHMM
Accuracy	PUZZLE	0.025^{***}	0.029^{***}	0.007^{***}	0.004^{*}	0.013^{***}	0.008^{***}	0.007^{**}	$0.005^{\dagger}$
	CLASSIFY	0.030^{***}	0.050^{***}	0.013^{**}	0.007^{**}	0.016^{***}	0.010^{**}	0.004	0.011^{**}
	TYPO	0.017^{*}	0.017^{*}	0.007	-0.001	-0.001	-0.009	-0.008	$0.009^{\dagger}$
F_1 score	PUZZLE	0.020^{***}	0.023^{***}	0.005^{***}	0.003^{*}	0.010^{***}	0.005^{**}	0.006^{***}	0.004^{*}
	CLASSIFY	0.049^{***}	0.061^{***}	0.018^{***}	0.011^{***}	0.016^{***}	0.013^{***}	0.010^{***}	0.021^{***}
	TYPO	0.023^{***}	0.023^{**}	0.011^{*}	0.000	0.001	-0.008	-0.006	0.011^{*}
Log loss	PUZZLE	-0.535^{***}	-0.055^{***}	-0.021^{***}	-0.003	-0.030^{***}	-0.028^{***}	-0.015^{***}	-0.003
	CLASSIFY	-0.719^{***}	-0.059^{***}	-0.018^{***}	-0.004	-0.023^{***}	-0.018^{***}	-0.010^{***}	-0.014^{***}
	TYPO	-0.586^{***}	-0.031^{***}	-0.041^{***}	-0.023^{***}	-0.046^{***}	-0.033^{***}	-0.100^{**}	-0.066^{***}

most always outperform the 2 baseline models on all evaluation metrics. For each dataset, the best-performing proposed model obtains a $2.2\%-8.2\%$ improvement on accuracy and F_1 score over the baseline models, and the log loss is also significantly decreased, especially compared to the running accuracy model. This suggests that when monetary interventions are provided in task sessions, it is necessary to explicitly model the impact of monetary interventions in order to characterize the temporal crowd work quality accurately and confidently.

Among all prediction models, the *random forests* model seems to outperform other models as its high performance has been consistently observed across all datasets. In fact, random forests is the best-performing prediction model according to all three metrics on the PUZZLE and CLASSIFY dataset, and it is also the best-performing model on the TYPO dataset according to the log loss value. Table 1 presents a detailed comparison between random forests and other models. In particular, given a specific metric, we have evaluated that metric 20 times for each prediction model as there are 20 random splits of training and testing data. Thus, for each model, we obtain a performance vector with 20 elements. To compare the performance of random forests with another model, we take the average for the corresponding performance vectors of both models and compute the difference in the average values (e.g. average accuracy of random forests − average accuracy of DARX), which are reported in Table 1. We further use *paired* t-test to examine whether these differences are statistically significant, and the results are noted as superscripts in Table 1. As we can see in the table, compared to other models, random forests almost always has a significantly higher accuracy (i.e. positive differences for accuracy), higher F_1 score (i.e. positive differences for F_1 score) and lower log loss (i.e. negative differences for log loss), and none of the differences in unexpected directions (e.g. negative differences for accuracy) are statistically significant. These results suggest that in practice, the random forests model gives high prediction performance for various types of tasks and thus is a good candidate model to use for requesters who are interested in making predictions on crowd work quality. We leave the problem of understanding why the random forests model is consistently accurate for future study.

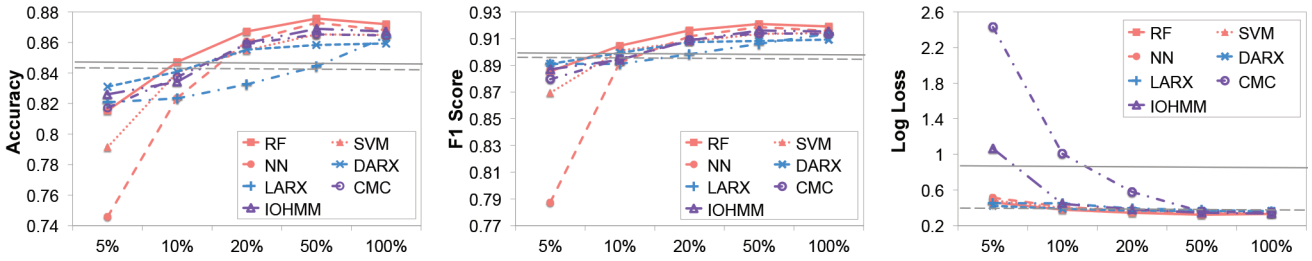
A closer look at the estimated random forest model further provides us with a few practical insights for understanding the role of monetary interventions on worker performance. On the one hand, we find that the *average performance* is the most important feature for predicting work quality in the current task; on the other hand, it is observed that among all intervention-related features (i.e. current intervention level, average intervention level, historical intervention levels, historical intervention changes), the *average intervention level* is the most informative one for the prediction.

Prediction with Limited Training Data

Next, we examine the performance of different models when the requester has limited training data to start with. To mimic the realistic scenario for the requester to obtain more training data over time, given a particular training dataset, we first randomly take 5% of the workers in it and train the models using *only* the data from these workers. After examining the performance of these models on the testing dataset, we pick another random 5% of the workers in the original training dataset who are *not* previously selected, and *combine* their data with the data from the first 5% workers to create a training dataset that consists of 10% of the workers in the original training dataset. Following the similar process, we construct two more training datasets, with 20% and 50% of the workers in the original training dataset, respectively[3].

Figure 2 illustrates the performance of different models on the PUZZLE dataset when the models are estimated from the 5%, 10%, 20%, 50% and the full training datasets. The performance of the prediction models improves as the amount of training data increases — With the training data from only 5% of the workers (i.e. 12 workers), all the 7 proposed models are actually inferior to the baseline LAR model according to all three metrics. Some models are especially sensitive to the size of the training dataset. For example, when the training data is very limited, SVM and NN suffers from a significantly lower accuracy and F_1 score, while CMC and IOHMM models have very high log loss values. On the

[3]For the TYPO dataset, we only construct two datasets with 20% and 50% of the workers in the original training dataset (the number of workers in the these two training datasets are 13 and 32, respectively) because the total number of workers in this dataset is relatively small.



(a) Accuracy (b) F_1 score (c) Log loss

Figure 2: Performance comparisons for all prediction models on the PUZZLE dataset when training data is limited. The solid and dashed gridlines are the performance references for the RA and LAR models, respectively (training datasets are not required for these two baseline models).

other hand, once the size of the training dataset has been increased to include 20% of all workers (i.e. 48 workers) in the original training dataset, almost all proposed models outperform both RA and LAR on all metrics. When the size of the training dataset further increases, while the prediction performance of different models keeps improving, the marginal benefit of extra training data also decreases. Importantly, we notice that even though the model is trained on only a fraction of the workers in the original training dataset, the random forests model still presents better prediction performance than other models in most cases, which suggests the robustness of this model against the limited training data. Similar results are also observed in the CLASSIFY and TYPO datasets.

Therefore, as a practical implication, a requester may consider to use the LAR model to predict crowd work quality in task sessions at the initial stage when they just start to recruit workers to work on their tasks. After collecting a small training dataset (e.g. a dataset of about 50 workers), the requester can switch to models that explicitly consider the impact of monetary interventions, especially the random forests model, to obtain more accurate predictions with higher confidence.

Predictions with Limited Ground Truth

Finally, we consider the scenario when the requester only has access to limited amount of ground truth. Ground truth information is quite valuable in crowdsourcing as in many cases, the requester will not be able to assess the work quality in a task without the ground truth. So far, we have assumed that the requester knows the ground truth to all his tasks hence he can evaluate the work quality for *every* task in a task session, and all the seven proposed models rely on the observation of past work quality (i.e. the sequence $\boldsymbol{y}$) when making predictions on work quality in the current task. To understand how the prediction performance of different models are influenced when this assumption is violated, that is, when the requester can only check the work quality for a limited number of tasks in the session, we conduct a new set of experiments.

In particular, given a specific split of training and testing data, prediction models are learned using the full training dataset as previously described[4]. When making predictions for workers in the testing dataset, we fix the first three tasks in each worker's session to be tasks with ground truth in order to obtain an initial record of the worker performance. Then, for the rest of the tasks in the session, we randomly select a certain portion (r) of them to be tasks with ground truth, hence work quality is only observable for these tasks. We vary this percentage, that is, $r \in \{0\%, 20\%, 40\%, 60\%, 80\%, 100\%\}$, and examine the performance of our prediction models in each of these cases when the ground truth is limited to different degrees.

For the simplicity of illustration, in this experiment, we focus on the two baseline models and three of the proposed models — RF, LARX and IOHMM, one from each category. For IOHMM, the lack of ground truth can be taken care of by simply updating the state belief in a different way when work quality is not observable[5]. For other models, we take a Monte Carlo approach to address the prediction problem: We maintain a set of $M = 100$ work quality sequences $Q = \{\boldsymbol{q_1}, \boldsymbol{q_2}, \cdots, \boldsymbol{q_M}\}$, where $\boldsymbol{q_m} = (q_m^1, q_m^2, \cdots, q_m^{l-1})(1 \leq m \leq M)$ is a sequence of "simulated" work quality for all the $l-1$ tasks provided to the worker so far. To forecast the worker's performance on the l-th task, we first make a prediction with each of the M work quality sequences and then take an average of all M predictions. That is, $Pr(y^l = 1) = \frac{1}{M}\sum_{m=1}^{M} p_m$, where p_m is the predicted probability of high-quality work on task l assuming that $\boldsymbol{q_m}$ is the observed work quality sequence for the past $l-1$ tasks. After the prediction, if the ground truth for task l is available hence the requester can actu-

[4]We assume that the requester can still evaluate the work quality on every task in the training dataset. This assumption is realistic, for example, if the requester bundles multiple tasks with ground truth into a single session and provide such task sessions to workers in the initial phase when the training dataset is collected. Models estimated from such training dataset can be used to predict work quality on tasks without ground truth when tasks with or without ground truth are similar (e.g. have similar difficulty levels).

[5]For example, when the order of the IOHMM is $p = 1$, the state belief is updated according to the formula $b_{l+1}(j) \propto \sum_{k=1}^{K} b_l(k) P_{tr}(j|k, a^l)$ if the requester doesn't have ground truth for the l-th task, rather than according to $b_{l+1}(j) \propto \sum_{k=1}^{K} b_l(k) P_{tr}(j|k, a^l) P_e(y^l|j, a^l)$.

ally decides the work quality y^l, we update $\boldsymbol{q_m}$ by setting $q_m^l = y^l$; otherwise, we sample a work quality $\hat{y}^l$ according to $Pr(\hat{y}^l = 1) = p_m$, and then update $\boldsymbol{q_m}$ as $(q_m^1, q_m^2, \cdots, q_m^{l-1}, \hat{y}^l)$.

Figure 3 plots for each of the 5 models, the change of average prediction performance as the amount of tasks with ground truth increases in the PUZZLE dataset. We find that RF, LARX and IOHMM models almost always make more accurate predictions with higher confidence compared to the baseline RA and LAR models. Among RF, LARX and IOHMM, the RF and IOHMM models are more robust when the requester has limited access to the ground truth information. In particular, the RF and IOHMM model outperforms the two baseline models as well as the LARX model in regardless of how small the fraction of tasks with ground truth is, and the prediction performance of RF and IOHMM when only 20% of the tasks has ground truth even exceeds the performance of the baseline models when the work quality is always observable for all tasks. Similar results are observed on other datasets, and they provide further supporting evidence for using the random forests model to predict crowd work quality under monetary interventions — it can not only make consistently accurate predictions for various types of tasks or given small set of training data, but also presents robust performance under limited supervision.

Conclusions

In this paper, we explore the potential of better characterizing the temporal pattern of crowd work quality by explicitly modeling the impact of external factors, like the provision of monetary interventions, on worker performance. We present a wide range of models from 3 categories, including supervised learning models, variants of autoregressive models and Markov models, and conduct an empirical comparison on the performance of these models in predicting crowd work quality under monetary interventions. Our results demonstrate that these proposed models indeed provide better predictions compared to baseline models. Furthermore, we identify the random forests model to be an excellent model for requesters to use to predict work quality in practice, as it presents consistently high performance for various types of tasks, and it is relatively robust against the size of the training dataset or the amount of available ground truth information.

There are many interesting future directions for this work. Firstly, as previous studies have shown that worker's behavioral traces in crowdsourcing tasks, such as how long they stay in a task and how they interact with the task interface, can be effective in predicting worker performance (Rzeszotarski and Kittur 2011; Sameki, Gurari, and Betke 2015), it will be an interesting future work to examine whether these behavioral traces can be integrated into the current models to further improve the prediction performance on crowd work quality under interventions. Secondly, the crowd work quality predictors presented in this paper can be incorporated into the general framework of crowdsourcing task design and be used to optimize interventions in crowdsourcing. For example, as discussed in Yin and Chen (2015), with an increased capability to predict work quality under monetary interventions in task sessions, the requester may provide monetary interventions to the right population who are more likely to react to the interventions, and at a better timing. Finally, extending these models to different contexts, such as to model worker engagement in reaction to the intervention messages from the crowdsourcing system, and further utilizing these models to guide the decisions on when and to whom to display intervention messages, will be another exciting future direction.

Acknowledgments

We thank the support of the National Science Foundation under grant CCF-1301976 on this work. Any opinions, findings, conclusions, or recommendations expressed here are those of the authors alone.

References

Bengio, Y., and Frasconi, P. 1995. An input output hmm architecture. *Advances in neural information processing systems* 427–434.

Bengio, Y., and Frasconi, P. 1996. Input-output hmms for sequence processing. *Neural Networks, IEEE Transactions on* 7(5):1231–1249.

Boser, B. E.; Guyon, I. M.; and Vapnik, V. N. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, 144–152. ACM.

Bragg, J., and Weld, D. S. 2016. Optimal testing for crowd workers. In *Proceedings of the 2016 International Conference on Autonomous Agents and Multiagent Systems*.

Cortes, C., and Vapnik, V. 1995. Support-vector networks. *Machine learning* 20(3):273–297.

Dai, P.; Rzeszotarski, J. M.; Paritosh, P.; and Chi, E. H. 2015. And now for something completely different: Improving crowdsourcing workflows with micro-diversions. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, 628–638. ACM.

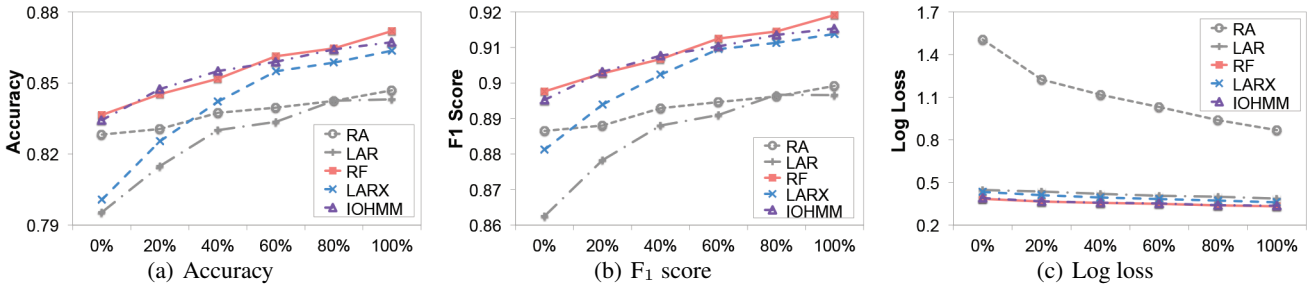
Donmez, P.; Carbonell, J. G.; and Schneider, J. G. 2010. A probabilistic framework to learn from multiple annotators with time-varying accuracy. In *SDM*, volume 2, 1. SIAM.

Dow, S.; Kulkarni, A.; Klemmer, S.; and Hartmann, B. 2012. Shepherding the crowd yields better work. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, 1013–1022. ACM.

Greengard, S. 2011. Following the crowd. *Communications of the ACM* 54(2):20–22.

Harris, C. 2011. You're hired! an examination of crowdsourcing incentive models in human resource tasks. In *Proceedings of the Workshop on Crowdsourcing for Search and Data Mining (CSDM) at the Fourth ACM International Conference on Web Search and Data Mining (WSDM)*, 15–18.

Ho, C.-J.; Slivkins, A.; Suri, S.; and Vaughan, J. W. 2015. Incentivizing high quality crowdwork. In *Proceedings of*



(a) Accuracy (b) F_1 score (c) Log loss

Figure 3: Performance comparisons for different prediction models on the PUZZLE dataset when ground truth is limited.

the 24th International Conference on World Wide Web, 419–429. International World Wide Web Conferences Steering Committee.

Ho, T. K. 1998. The random subspace method for constructing decision forests. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 20(8):832–844.

Hornik, K.; Stinchcombe, M.; and White, H. 1989. Multilayer feedforward networks are universal approximators. *Neural networks* 2(5):359–366.

Huang, S.-W., and Fu, W.-T. 2013. Enhancing reliability using peer consistency evaluation in human computation. In *Proceedings of the 2013 conference on Computer supported cooperative work*, 639–648. ACM.

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

Jung, H. J., and Lease, M. 2015a. A discriminative approach to predicting assessor accuracy. In *Advances in Information Retrieval*. Springer. 159–171.

Jung, H. J., and Lease, M. 2015b. Modeling temporal crowd work quality with limited supervision. In *Third AAAI Conference on Human Computation and Crowdsourcing*.

Jung, H. J.; Park, Y.; and Lease, M. 2014. Predicting next label quality: A time-series model of crowdwork. In *Second AAAI Conference on Human Computation and Crowdsourcing*.

Karger, D. R.; Oh, S.; and Shah, D. 2011. Iterative learning for reliable crowdsourcing systems. In *Advances in neural information processing systems*, 1953–1961.

Lazebnik, S.; Schmid, C.; and Ponce, J. 2004. Semi-local affine parts for object recognition. In *British Machine Vision Conference (BMVC'04)*, 779–788. The British Machine Vision Association (BMVA).

Lin, C. H.; Mausam; and Weld, D. 2012. Dynamically switching between synergistic workflows for crowdsourcing. In *In Proceedings of the 26th AAAI Conference on Artificial Intelligence, AAAI12*. Citeseer.

Ljung, L. 1998. *System identification*. Springer.

Mao, A.; Kamar, E.; and Horvitz, E. 2013. Why stop now? predicting worker engagement in online crowdsourcing. In *First AAAI Conference on Human Computation and Crowdsourcing*.

Mason, W., and Watts, D. J. 2010. Financial incentives and the performance of crowds. *ACM SigKDD Explorations Newsletter* 11(2):100–108.

Park, Y.; Carvalho, C.; and Ghosh, J. 2014. Lamore: A stable, scalable approach to latent vector autoregressive modeling of categorical time series. In *AISTATS*, 733–742.

Raykar, V. C., and Yu, S. 2012. Eliminating spammers and ranking annotators for crowdsourced labeling tasks. *Journal of Machine Learning Research* 13:491–518.

Rzeszotarski, J. M., and Kittur, A. 2011. 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*, 13–22. ACM.

Sameki, M.; Gurari, D.; and Betke, M. 2015. Predicting quality of crowdsourced image segmentations from crowd behavior. In *Third AAAI Conference on Human Computation and Crowdsourcing*.

Segal, A.; Gal, Y.; Kamar, E.; Horvitz, E.; Bowyer, A.; and Miller, G. 2016. Intervention strategies for increasing engagement in crowdsourcing: Platform, predictions, and experiments. In *Proceedings of the 25th International Conference on Artificial Intelligence*.

Shaw, A. D.; Horton, J. J.; and Chen, D. L. 2011. Designing incentives for inexpert human raters. In *Proceedings of the ACM 2011 conference on Computer supported cooperative work*, 275–284. ACM.

Whitehill, J.; Wu, T.-f.; Bergsma, J.; Movellan, J. R.; and Ruvolo, P. L. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in neural information processing systems*, 2035–2043.

Yin, M., and Chen, Y. 2015. Bonus or not? learn to reward in crowdsourcing. In *Proceedings of the 24th International Conference on Artificial Intelligence*, 201–207. AAAI Press.

Yin, M.; Chen, Y.; and Sun, Y.-A. 2013. The effects of performance-contingent financial incentives in online labor markets. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*.