Dynamic Task Allocation Algorithm for Hiring Workers that Learn

Shengying Pan
University of Waterloo
Canada
s5pan@uwaterloo.ca

Kate Larson
University of Waterloo
Canada
kate.larson@uwaterloo.ca

Josh Bradshaw
University of Waterloo
Canada
jabradsh@uwaterloo.ca

Edith Law
University of Waterloo
Canada
edith.law@uwaterloo.ca

Abstract

The automation of hiring decisions is a well-studied topic in crowdsourcing. Existing hiring algorithms make a common assumption—that each worker has a level of task competence that is static and does not vary over time. In this work, we explore the question of how to hire workers who can learn over time. Using a medical time series classification task as a case study, we conducted experiments to show that workers' performance does improve with experience and that it is possible to model and predict their learning rate. Furthermore, we propose a dynamic hiring mechanism that accounts for workers' learning potential. Through both simulation and real-world crowdsourcing data, we show that our hiring procedure can lead to high-accuracy outcomes at lower cost compared to other mechanisms.

1 Introduction

Crowdsourcing has become a prevalent tool for outsourcing tasks with varying complexity. One of the most common type of crowdsourcing tasks is *consensus tasks*, where workers are asked to provide opinions which are then aggregated to predict the answer to a question. From the requester's perspective, the objective is to hire as few workers as possible to save on the cost while still generating accurate outputs from the aggregated opinions.

There exists a rich body of research that explored mechanisms for hiring the best workers. In most cases, such mechanisms assume that the quality of workers is fixed. In reality, workers can learn from experience and improve over time. Mechanisms that only select the top workers may end up with mediocre workers, by ignoring incoming workers with low initial quality but high learning potential; such a system is not optimal in the long run.

In this paper, we propose to model the learning process of crowd workers using hyperbolic learning curves, to estimate both their current quality as well as projected future improvement. Our work makes two contributions. First, through experimentation, we demonstrate that workers performing a complex consensus task—identifying sleep spindles, a particular kind of EEG pattern—do learn and improve over time.

Second, we introduce a dynamic hiring mechanism which allocates tasks to workers, not based on their current quality, but based on their learning potential. Results, from both simulation and crowdsourcing experiments involving the sleep spindle detection task, show that our algorithm can save cost, while achieving high accuracy.

1.1 Related Work

There are several strategies for improving the quality of consensus tasks in the absence of ground truth—one can hire more workers to redundantly perform the same task [Sheng *et al.*, 2008; Lin *et al.*, 2014], intelligently weigh workers based on their inferred expertise [Donmez *et al.*, 2009; Raykar *et al.*, 2010; Welinder *et al.*, 2010], or hire a small group of top quality workers [Zhao *et al.*, 2013; Li *et al.*, 2014; Li and Liu, 2015; Carvalho *et al.*, 2016]. Our approach follows the third strategy, except that our hiring algorithm uses not only current performance, but also learning potentials, as the criteria for choosing the "top" workers.

Prior work on hiring algorithms are centered on *exploration-exploitation* strategies that use some portion of the task budget to learn the quality of the available workers, then use the remaining budget to hire the best worker. For example, Tran-Thanh *et al.* introduces a hiring algorithm based on a variation of the multi-armed bandit model [Tran-Thanh *et al.*, 2014]; Donmez *et al.* presents a sequential Bayesian estimation algorithm that continuously tracks and selects the best labelers over time [Donmez *et al.*, 2010]. Closest to our approach is Kamar *et al.* [Kamar *et al.*, 2012; 2013; Kamar and Horvitz, 2015], which use machine learning to model agents' behaviour doing consensus tasks, and use this model to predict a candidate worker's behaviour. They model the series of hiring decisions as a Markov Decision Process, where the rewards are the system's belief in the correctness of the aggregate prediction from the hired workers minus the cost to hire those workers. Finally, there are also recent approaches for modeling the time-varying performance of workers [Jung *et al.*, 2014]. In all these approaches, it is assumed that workers have a level of quality that may fluctuate between tasks, but remains relatively constant over time.

2 Do Workers Learn?

In the first study, our goal is to establish whether workers exhibit learning behavior while performing consensus tasks,
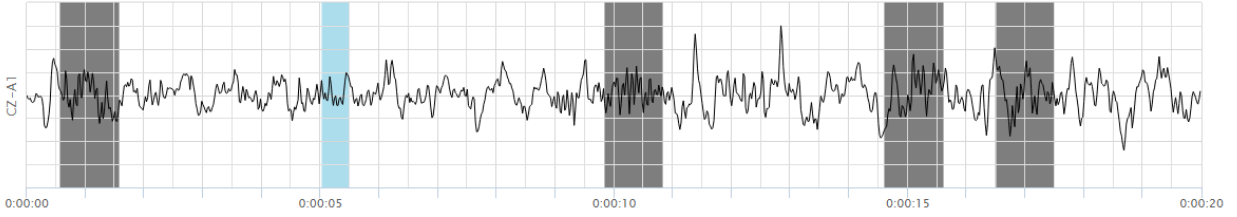
Figure 1: sleep spindles detection task with feedback to worker

and whether we can model and predict their improvement using learning curves. In industrial engineering, prior work has shown that the quality of workers improve as they complete repetitive tasks [Adler and Clark, 1991; Vits and Gelders, 2002]. Such improvements are often modeled using learning curves—a mathematical description of worker's performance for repetitive tasks [Fioretti, 2007]. While numerous forms of learning curves have been proposed, we use the *hyperbolic model* in this work since it was designed to measure and predict each worker's percentage of correctly completed tasks in production scenarios [Mazur and Hastie, 1978]. This can be translated directly to our setting of crowdsourcing consensus tasks as the percentage of correctly predicted or labeled tasks. Additionally, there is strong evidence that the hyperbolic curve model is a well studied and validated model which outperforms many other models in terms of efficiency, stability, and robustness [Nembhard and Uzumeri, 2000; Anzanello and Fogliatto, 2007]. Our experimental results here provide support for this argument.

Let w be a worker with learning speed r_w and prior knowledge p_w. Then, the hyperbolic learning curve model states that the percentage of correct predictions worker w has made up till the x'th task (*i.e.* its *cumulative quality*) is defined as

$$Q_w(x) = \frac{x + p_w}{x + p_w + r_w}. \qquad (1)$$

Given $Q_w(x)$, we can calculate $q_w(x)$, the worker's quality (the probability of making a correct prediction) at task x as:

$$q_w(x) = xQ_w(x) - (x-1)Q_w(x-1). \qquad (2)$$

If r_w and p_w were known in advance, then computing $Q_w(x)$ and $q_w(x)$ would be straightforward. Instead, we approximate $Q_w(x)$ by estimating the number of correct answers each worker has provided so far (*e.g.*, via comparison against ground truth data). To learn a model for $Q_w(x)$, we map a linear model to the data:

$$Z_w(x) = \alpha_w x + \beta_w$$

where $Z_w(x) = \frac{1}{1-Q_w(x)}$, $\alpha_w = \frac{1}{r_w}$ and $\beta_w = \frac{p_w}{r_w} + 1$, and use linear-regression tools to estimate the parameters of $Z_w(x)$, and thus estimate $Q_w(x)$.

Modeling workers' learning curves has two advantages. First, we can use the estimated r_w's to rank workers by learning speed, and select the most promising ones to train. Furthermore, with $Q_w(x)$ and $q_w(x)$, we can estimate the probability a worker will make a correct prediction in future tasks, and use this information in our hiring decisions.

2.1 Crowdsourcing Study

To assess the goodness of fit of the learning curve model, we hired crowdworkers on Amazon Mechanical Turk to perform sleep spindle identification tasks [Warby *et al.*, 2014]. A sleep spindle is a discrete, intermittent pattern appearing on sleep-study EEG recordings, which neurologists use to identify particular sleep stages. A sleep spindle is identified based on its waxing/waning shape (*i.e.*, like a diamond or football), frequency (*i.e.*, oscillate at approximately 12-15 cycles per second), and duration (*i.e.*, mostly between 0.5 to 1.0 seconds in length), and amplitude (usually slightly taller than the waves around it). The task of identifying sleep spindles is a fitting case study for our setting—it is straightforward enough that the crowdworkers are able to understand how to proceed just by reading the instructions, but the task is challenging enough that one would expect workers' performance to be low at the onset and improve with training and feedback.

After reviewing a brief instructions page, workers were provided with a sequence of 20 windows of EEG recording. We asked workers to identify all the sleep spindles in the recording by clicking and dragging boxes around them. After the worker submits each window, the system provides feedback, revealing the actual locations of the sleep spindles before showing the next window. All workers are given the exact same sequence of 54 windows and each window contains at least one sleep spindle. We removed workers who did not complete all 54 windows and workers who spent less than 10 minutes completing all tasks, which is less than 25% of the average time required to complete the task. After filtering, we have 10 workers whose data we used for our analysis. All the EEG recordings and ground truth sleep spindle identifications used in our experiment come from Devuyst's DREAMS Sleep Spindle Database [Devuyst *et al.*, 2011].

The primary goal of this study is to assess whether the crowd *as a whole* is improving with experience, and the feasibility of using learning curves to model their improvement. For each annotation, we considered it to be correct if the bounding box overlapped with a ground truth identification. We measured each worker's quality using common metrics, such as precision, recall, F_1 and F_2 scores. We then computed a cumulative quality measure by aggregating workers' performance on a per window basis. Cumulative quality at the i-th task is computed by aggregating all reported sleep spindles from all 10 workers up to window i. This cumulative quality metric enables us to apply the learning curve model and avoid the sometimes drastic fluctuation in workers' task-to-task performance, thereby better modeling the general per-
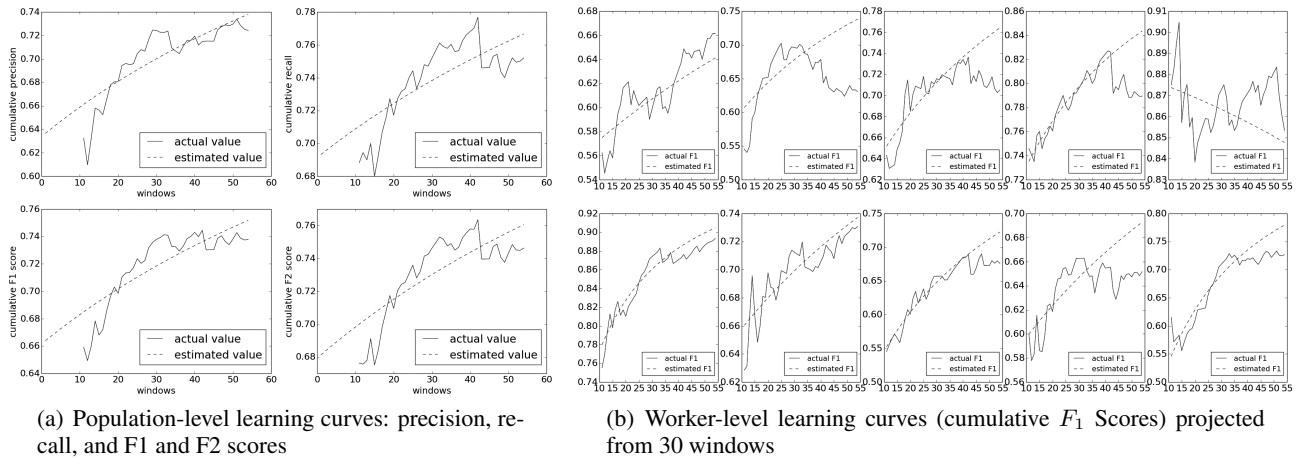
(a) Population-level learning curves: precision, recall, and F1 and F2 scores

(b) Worker-level learning curves (cumulative F_1 Scores) projected from 30 windows

Figure 2: fitting learning curves to worker performance

measurement	r	p	standard error	p-value
precision	87	190	0.0019	3.03×10^{-7}
recall	54	109	0.0024	1.90×10^{-9}
F_1	67	140	0.0019	4.04×10^{-10}
F_2	58	120	0.0021	5.16×10^{-10}

Table 1: Estimated population-level learning speed, prior knowledge, standard error and P-value

formance trend over time.

The estimated learning speed r, prior knowledge p, and associated standard error and p-values from the linear regression test are shown in Table 1. The population-level and worker-level learning curves (in Figure 2(a) and Figure 2(b)) show that cumulative quality (solid curves) has a noticeable upward trend and has a significant relationship ($p << 0.05$) with our proposed hyperbolic learning curve (dashed curves). These results give us insight into the difficulty level of the tasks, *e.g.* how many tasks in general are needed for a worker with no prior knowledge to reach 50% quality, and the overall quality of the worker population, *e.g.* workers in average have a quality score of around 70% at the beginning.

3 A Dynamic Hiring Procedure

In this section, we show how it is possible to incorporate information about workers' learning into the hiring process, allowing the system to balance the quality of task output, the cost of hiring workers, and the future benefit derived from assigning tasks to workers for training purposes.

3.1 A Markov Decision Process for Worker Hiring

Let W be the set of workers, where the quality q_w of each worker $w \in W$ is defined by his learning speed r_w, prior knowledge level p_w and number of completed tasks x_w. Similar to other researchers (see, for example [Lin *et al.*, 2012; Dai *et al.*, 2010]), we use a decision-theoretic framework to model the hiring problem for a single task. We define a finite-horizon Markov Decision Process (MDP) as $\langle l, S, \mathbb{A}, T, R \rangle$ where l is the task horizon (the maximum number of workers

the system will hire for a single task), S is the state space, $\mathbb{A}$ is the action set, T is the transition probability function, and R is the reward function.

States: A state, $s_t \in S$ is defined as a set of workers hired until time t, along with an opinion a_w from each worker w. That is $s_t = \{(w_1, a_1), \ldots, (w_t, a_t)\} = \{(w_i, a_i)_{i=1}^t\}$. For ease of explanation we assume that the tasks are structured so that there are only two opinions (for example, *yes* or *no*); however our model easily generalizes to situations where the set of possible opinions is larger.

Actions: At each state s_t, the system can take one of two actions—terminate hiring and return the workers' aggregated opinions to the task owner ($\neg H$), or select a new worker, w_i, to hire from the worker pool (H_{w_i}). That is, $\mathbb{A} = \{\neg H\} \cup \{H_w | w \in W\}$.

If action $\neg H$ is taken in state $s_t = \{(w_i, a_i)_{i=1}^t\}$, the workers' opinions are aggregated as follows: Assuming that workers' opinions are conditionally independent given the correct classification $\hat{a}$, the system's *belief*, b, that classification a is correct is

$$b(a|\{a_i\}_1^t) = Pr(\hat{a} = a|\{a_i\}_1^t) \quad (3)$$

$$= \lambda_a Pr(\{a_i\}_1^t | a) Pr(a)$$

$$= \lambda_a Pr(a) \prod_{i=1}^t Pr(a_i|a).$$

where λ_a is a normalizing factor and $Pr(a)$ is the prior probability. Since, for each worker, we know their current *quality*, q_w, we have

$$Pr(a_i|a) = \begin{cases} q_{w_i} & \text{if } a_i = a \\ 1 - q_{w_i} & \text{if } a_i \neq a \end{cases} \quad (4)$$

The system returns the classification

$$a^* = \arg\max_a b(a|\{a_i\}_1^t)$$

and *updates* its knowledge about the workers. In particular, for each worker w hired, $x_w \leftarrow x_w + 1$ which induces changes in q_w. When asked to hire workers for a new task, the system uses this updated worker-population information.

Transitions: If, in state $s_t = \{(w_i, a_i)_{i=1}^t\}$, action H_w is selected, then the system transitions to state $s_{t+1} = \{(w_i, a_i)_{i=1}^t \cup (w, a_w)\}$ with probability $T(s_t, H_w, s_{t+1}) = Pr(a_w | \{a_i\}_{i=1}^t)$. Letting a represent the true classification of the task, and assuming that workers' responses are conditionally independent given the true classification,

$$T(s_t, H_w, s_{t+1}) = \lambda \sum_a Pr(a_w|a) \prod_{i=1}^t Pr(a_i|a) Pr(a) \quad (5)$$

where λ is the normalizing factor, and $Pr(a_i|a)$ is defined by Equation 4.

Rewards: There are two types of rewards in the system: the *aggregation reward* and the *training reward*. These rewards, together, allow the system to strike a balance between the goal of hiring workers so as to improve the quality of their aggregated opinions and hiring workers so as to provide them with training to improve their quality for future tasks.

The *aggregation reward*, $R(s, \neg H)$, assuming that the system is in state s and takes action $\neg H$, is defined by

$$R(s, \neg H) = \begin{cases} \beta(2^{b(a|\{a_i\}_{i=1}^t)} - 1) & \text{if } b(a|\{a_i\}_{i=1}^t) \geq b_t \\ 0 & \text{otherwise} \end{cases}$$

where b_t is a threshold parameter so that only aggregated opinions with high enough belief are rewarded and β is a weighting term which allows the system to tune the reward of an accurate prediction to the actual cost of hiring a worker. In particular, the aggregation reward is an increasing function of the belief the system has in the correctness of the aggregated response from the workers.

The *training reward*, $R(s, H_w)$, is awarded when the system is in state s and takes action H_w (i.e., hires a worker w). In its most basic form, we have $R(s, H_w) = -c$ where c is the cost of hiring a single worker. However, we also find it useful to use reward shaping (see, for example [Ng *et al.*, 1999]) and allow part of the system reward to directly incorporate information about the change in worker experience and quality. In particular, by hiring worker w, the worker gains more experience and thus its quality changes, as described by its learning curve. If the worker had previously completed x_w tasks, then by being hired for an additional task we define the change in quality as

$$q_\Delta(w) = (2^{q_w(x_w+1)} - 1) - (2^{q_w(x_w)} - 1).$$

The associated *training reward* is

$$R(s, H_w) = \gamma q_\Delta(w) - c$$

where, again, c is the cost of hiring a single worker, and γ is a weighting parameter that allows the system to tune the training reward to the actual cost of hiring a worker.

While we define specific rewards for the system, they are not necessarily unique as the appropriate reward structure may depend on the specific domain. However, we argue that our proposed reward functions are desirable as they are convex functions and so provide higher rewards for *more challenging* aggregation and quality improvement.

Solving the MDP

Given $MDP = \langle l, S, \mathbb{A}, T, R \rangle$, the optimal policy π^* specifies an action for each state so that the system utility is maximized. In particular, an optimal policy π^* with value function V^{π^*} satisfies the Bellman equation

$$V^{\pi^*}(s_t) = \max_{\alpha \in \mathbb{A}} [R(s_t, \alpha) + \sum_{s_{t+1}} T(s_t, \alpha, s_{t+1}) V^{\pi^*}(s_{t+1})].$$

Alternatively, since each action H_w is an information-gathering action, we can reformulate the hiring problem as a *value of information (VOI)* problem where the *VOI* at state $s_t = \{(w_i, a_i)_{i=1}^t\}$, given worker w is hired, is the expected utility of hiring w rather than stopping and aggregating the opinions $\{a_i\}_{i=1}^t$. In particular,

$$VOI(s_t, w) = R(s_t, H_w) + \sum_{s_{t+1}} T(s_t, H_w, s_{t+1}) V^{\pi^*}(s_{t+1}) - R(s_t, \neg H)$$

and

$$VOI(s_t) = \max_w VOI(s_t, w).$$

If, at state s_t, $VOI(s_t) > 0$ then the system is best off hiring worker $w^* = \arg\max_{w \in W} VOI(s_t, w)$. Otherwise, the system is best off terminating the hiring process and aggregating the workers' opinions and returning the task classification.

Since computing the expected value of information for sequences of observations under uncertainty is intractable [Heckerman *et al.*, 1993], researchers have proposed using sampling-based methods (for example, [Kearns *et al.*, 2002]). In particular, Kamar and Horvitz proposed the *MC-VOI* algorithm, a Monte-Carlo planning algorithm that explores the search space by sampling possible hiring paths, and then evaluates the rewards of either hiring a worker versus stopping and aggregating the collected responses for all states along the sampled path [Kamar and Horvitz, 2013].

We use the *MC-VOI* algorithm, modifying it to handle our larger search space and richer worker population. In the rest of this section we describe our modifications to *MC-VOI*, and direct the reader to the original paper introducing *MC-VOI* for full details of that algorithm [Kamar and Horvitz, 2013]. There are two main components to the MC-VOI algorithm, *sampling* and *evaluation*.

Sampling Phase: The sampling phase starts with the initial state where no workers have been hired and proceeds to hire workers until the maximum number is reached (l). If state s_t was sampled at step t and then action H_w was taken, the probability state s_{t+1} is sampled is defined by Equation 5. Each time a new state is encountered, a new node is added to the search tree. Once l workers have been hired, the aggregated response of the system is determined by Equation 3.

While *MC-VOI* assumes all workers are interchangeable, in our setting this is not the case. Ideally one should sample all available workers at any hiring state, but as the number of workers increases, the sampling tree grows exponentially and this becomes infeasible. Instead, we propose a *priority score* for workers that balances their immediate quality with future

expected quality, and sample workers in proportion to their score. Recall that each worker w has completed x_w tasks before the task of interest, and thus has quality $q_w(x_w)$. If the worker was allowed to complete an additional n tasks, then its projected quality is $q_w(x_w + n)$. We define the *score* of worker w as

$$S_w(x_w, n) = \delta_{\text{now}}q_w(x_w) + \delta_{\text{future}}q_w(x_w + n)$$

where δ_{now} and δ_{future} are weights that allow us to balance current and future quality. We sample worker w according to probability

$$Pr(w) = \frac{S_w(x_w, n)}{\sum_{w' \in W} S_{w'}(x_{w'}, n)},$$

and the worker's response, a_w is sampled based on $q_w(x_w)$.

Evaluation Phase: The evaluation phase updates the utility of each sampled state from bottom to top once sampling is done. For any state, s_l, in which there are l hired workers, $V(s_l) = R(s_l, \neg H)$. For any state s_t with $t < l$, we compute the estimated value of information of hiring worker w. Define $g(s_t, a_w)$ to be the state s_{t+1} that arises if worker w is hired in state s_t and provides opinion a_w. Then,

$$\begin{aligned} VOI^*(s_t, w) = &R(s_t, H_w) \\ &+ \sum_{a_w} T'(s_t, H_w, g(s_t, a_w))V(g(s_t, a_w)) \\ &- R(s_t, \neg H) \end{aligned}$$

where

$$T'(s_t, H_w, g(s_t, a_w)) = \frac{\text{\# smpls with } s_t, \text{ hiring } w \text{ with } a_w}{\text{\# smpls with } s_t}$$

is the probability of transitioning from state s_t to $g(s_t, a_w)$ based on the observations from the sampling procedure. If $VOI^*(s_t, w) < 0$ for all w then the optimal action is $\neg H$, and $V(s_t) = R(s_t, \neg H)$. Otherwise,

$$\begin{aligned} V(s_t) = \max_w [&R(s_t, H_w) \\ &+ \sum_{a_w} T'(s_t, H_w, g(s_t, a_w))V(g(s_t, a_w))]. \end{aligned}$$

and the action to be taken in state s_t is set to H_w.

4 Results

We compare our dynamic hiring procedure to two baselines— **RandomK**, which randomly picks k available workers, and **TopK**, which ranks all available crowd workers by their observed quality and picks the top k workers to work on the next task. For both baselines, we use majority voting to aggregate opinions to derive a single prediction. We compared the performance of the algorithms based on three metrics: n_c (number of correct answers), c_h (cost of hiring) and c_t (cost of training). c_t is a relative monetary measurement where we fix the hiring cost of one worker to perform one task to be 1. Finally, for all experiments, when scoring workers we set $\delta_{\text{now}} = \delta_{\text{future}} = 0.5$ and n to be the number of tasks remaining.

Simulation Results
We simulated 100 workers whose performance follows a parameterized hyperbolic learning curve model, with learning speed and prior knowledge drawn from truncated normal distribution $R(x) = f(x; 50, 5, 0, \infty)$ and $P(x) = f(x; 80, 5, 0, \infty)$. These parameters are based on the average values observed from sleep spindle detection workers during the crowdsourcing study. We then created 1000 binary tasks where both outcomes are equally likely to happen, ran each hiring algorithm 30 times (keeping all settings constant), and averaged the results (shown in Table 2 and Figure 3).

	uniform			heterogeneous		
Mechanism	n_c	c_h	c_t	n_c	c_h	c_t
RandomK	771	3000	0	606	3000	0
TopK	985	3000	2000	807	3000	2000
DynamicHiring	981	1522	800	931	1901	948

Table 2: Simulation results for worker population with uniform versus heterogeneous learning rates: number of correct answers, hiring cost and training cost

First, we simulated a population of workers with relatively uniform learning rates. Results (in Figure 3(a)) show that the performance of **RandomK** is improving over time in general, because each worker is given an opportunity to learn by performing some tasks. In contrast, **TopK** performs well since in our setting all workers share a similar learning speed which is relatively fast. No matter which workers were picked, their performance improved to produce very accurate predictions for the system. Our proposed **DynamicHiring** saves cost (c_h=1522, c_t=800) on hiring while maintaining high overall accuracy compared to **TopK** (c_h=3000, c_t=2000) — which is a 49.3% reduction of hiring cost, and 53.6% reduction taking hiring *and* training costs into account.

Next, we created a *heterogeneous* worker population, where the majority of workers are very slow learners with a better starting quality, while a small group of workers starts poorly but are faster learners and can outperform others quickly. In particular, we draw 80 slow learners from $R(x) = f(x; 600, 5, 0, \infty)$ and $P(x) = f(x; 800, 10, 0, \infty)$, with an average starting quality of 57%, and 20 fast learners from $R(x) = f(x; 60, 5, 0, \infty)$ and $P(x) = f(x; 40, 5, 0, \infty)$ with an average starting quality of 40%. This is a challenging situation for **TopK**, which may be easily misled to hire slow learners with better starting quality than to take advantage of faster learners and their better performance in the future. Our goal is to demonstrate that **DynamicHiring**, which takes into account learning rates of individual workers, is adaptive and robust enough to handle this challenging scenario.

Results (in Table 2 and Figure 3(b)) show that due to the small number of high potential workers, it takes **DynamicHiring** quite a while (i.e., 200 tasks) to locate and hire these workers. The fluctuation towards the end is due to the algorithm constantly exploring unknown workers, who have a lower initial quality but are recognized by the system as potential fast learners. By taking advantage of fast learners, **DynamicHiring** (n_c=931) outperformed **TopK** (n_c=807) by a huge margin, with a 15.4% improvement in quality. More im-

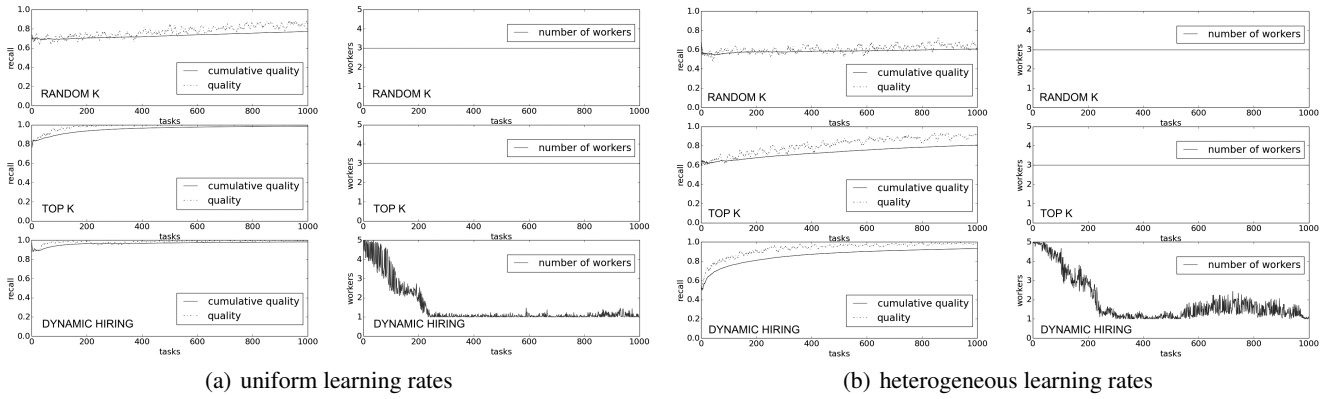(a) uniform learning rates   (b) heterogeneous learning rates

Figure 3: Simulation results: cumulative quality and number of hired workers

portantly, the hiring-cost savings remain significant (36.6% reduction, or 43% taking into account the cost of training tasks).

Overall, our simulations show that, for both worker populations with uniform and heterogeneous learning rates, our proposed dynamic hiring mechanism is able to yield similar performance to **TopK**, while providing significant savings on hiring and training costs.

Sleep Spindle Detection

Finally, we tested our dynamic hiring algorithm on the sleep spindle detection task. We assigned the first 20 tasks/spindles as training tasks; that is, **TopK** sampled all workers for 20 tasks first before hiring for remaining tasks, and **DynamicHiring** estimated the learning curve model based on worker's tutorial session during these 20 tasks. We retained only workers who annotated all 54 windows, which leaves 15 workers in total. We set $k = 3$ for both **RandomK** and **TopK** so they are not hiring excessive workers and there is no need for any tie breaking. We set the horizon of **DynamicHiring**, l, equal to 5 so it can explore a bit more at the beginning. For the reward functions we set $\beta = 7.0$, $b_t = 0.85$, and $\gamma = 100.0$. We ran the experiments 30 times and reported the average performance. After removing the first 20 sleep spindles, there are a total of 81 tasks left for testing.

Mechanism	n_c	c_h	c_t
RandomK	61	243	0
TopK	66	243	300
DynamicHiring	64	88	120

Table 3: Spindle detection task results: number of correct answers, hiring cost and training cost

Results (in Table 3 and Figure 4) show that **DynamicHiring**, **TopK** and **RandomK** achieved similar quality. However, **DynamicHiring** (c_h=208) provides a 62% reduction in terms of hiring cost compared to **TopK** (c_h=543); in fact, it costs even less than **RandomK** which has no training tasks. In other words, for this non-trivial classification task, our dynamic hiring procedure provided significant cost saving without losing much on performance, demonstrating the feasibility of our approach in real-world crowdsourcing settings.
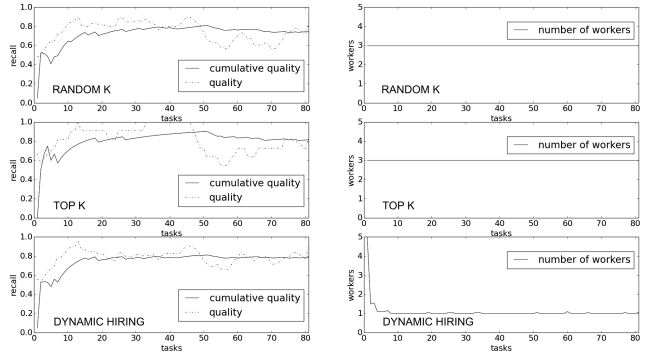

Figure 4: Spindle detection task results: cumulative quality versus number of hired workers

5 Conclusion

In this work, we demonstrated that, for certain types of tasks, crowd workers learn from experience and their quality of work may improve over time. We demonstrated, through 15 independent case studies, that it is possible to model each worker's learning curve and presented a decision-theoretic hiring model that accounts for the learning processes of the workers. Both simulation-based and experimental results illustrate that our model and approach are feasible—our hiring algorithm reduces hiring costs and provides competitive performance in terms of accuracy with other commonly used hiring mechanisms.

There are a number of future directions for this line of research. First, our dataset for the experimental validation was small; in future work, we aim to test our model on larger real-world crowdsourcing problems. Our model makes the unrealistic assumption that workers are always available; thus, a promising next step is to extend our model and algorithms to settings where workers may enter and leave the platform, leading to interesting dynamics as the system tries to find the right balance between greedily aggregating answers from currently available workers and training promising workers whose future availability is uncertain. Finally, we are interested in expanding our model beyond consensus tasks to handle more complicated (e.g., hierarchical) tasks.

References

[Adler and Clark, 1991] P. S. Adler and K. B. Clark. Behind the learning curve:A sketch of the learning process. *Management Science*, 37(3):267–281, 1991.

[Anzanello and Fogliatto, 2007] M. J. Anzanello and F. S. Fogliatto. Learning curve modelling of work assignment in mass customized assembly line. *International Journal of Production Research*, 45(13):2919–2938, 2007.

[Carvalho et al., 2016] A. Carvalho, S. Dimitrov, and K. Larson. How many crowdsourced workers should a requester hire? *Annals of Mathematics and Artificial Intelligence*, 2016.

[Dai et al., 2010] P. Dai, Mausam, and D. S. Weld. Decision-theoretic control for crowdsourced workflows. In *AAAI'10*, 2010.

[Devuyst et al., 2011] S. Devuyst, T. Dutoit, P. Stenuit, and M. Kerkhofs. Automatic sleep spindles detection - overview and development of a standard proposal assessment method. *In Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1713–1716, 2011.

[Donmez et al., 2009] P. Donmez, J. Carbonell, and J. Schneider. Efficiently learning the accuracy of labeling sources for selective sampling. In *KDD'09*, pages 259–268, 2009.

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

[Fioretti, 2007] G. Fioretti. The organizational learning curve. *European Journal of Operational Research*, 177(3):1375–1384, 2007.

[Heckerman et al., 1993] D. Heckerman, E. Horvitz, and B. Middleton. An approximate nonmyopic computation for value of information. *TPAMI*, 15(3):292–298, 1993.

[Jung et al., 2014] H. J. Jung, Y. Park, and M. Lease. Predicting next label quality: A time-series model of crowdwork. In *HCOMP'14*, pages 1–9, 2014.

[Kamar and Horvitz, 2013] E. Kamar and E. Horvitz. Light at the end of the tunnel: A Monte Carlo approach to computing value of information. In *AAMAS'13*, pages 571–578, 2013.

[Kamar and Horvitz, 2015] E. Kamar and E. Horvitz. Planning for crowdsourcing hierarchical tasks. In *AAMAS'15*, pages 1191–1199, 2015.

[Kamar et al., 2012] E. Kamar, S. Hacker, and E. Horvitz. Combining human and machine intelligence in large-scale crowdsourcing. In *AAMAS'12*, pages 467–474, 2012.

[Kamar et al., 2013] E. Kamar, A. Kapoor, and E. Horvitz. Lifelong learning for acquiring the wisdom of the crowd. In *IJCAI'13*, pages 2313–2320, 2013.

[Kearns et al., 2002] M. Kearns, Y. Mansour, and A. Ng. A sparse sampling algorithm for near-optimal planning in large Markov decision processes. *Machine Learning*, 49(2-3):193–208, 2002.

[Li and Liu, 2015] H. Li and Q. Liu. Cheaper and better: Selecting good workers for crowdsourcing. In *HCOMP'15*, pages 20–21, 2015.

[Li et al., 2014] H. Li, B. Zhao, and A. Fuxman. The wisdom of minority: Discovering and targeting the right group of workers for crowdsourcing. In *WWW'14*, pages 165–175, 2014.

[Lin et al., 2012] C. H. Lin, Mausam, and D. S. Weld. Dynamically swithcing between synergistic workflows for crowdsourcing. In *AAAI'12*, 2012.

[Lin et al., 2014] C.H. Lin, Mausam, and D. S. Weld. To re(label), or not to re(label). In *HCOMP'14*, pages 1–8, 2014.

[Mazur and Hastie, 1978] J. E. Mazur and R. Hastie. Learning as accumulation: A reexamination of the learning curve. *Psychological Bulletin*, 85(6):1256–1274, 1978.

[Nembhard and Uzumeri, 2000] D. A. Nembhard and M. V. Uzumeri. An individual-based description of learning within an organization. *IEEE Transactions on Engineering Management*, 47(3):370–378, 2000.

[Ng et al., 1999] A. Ng, D. Harada, and S. Russell. Policy invariance under reward transformation: Theory and application to reward shaping. In *ICML'99*, pages 278–287, 1999.

[Raykar et al., 2010] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from crowds. *J. Mach. Learn. Res.*, 11:1297–1322, August 2010.

[Sheng et al., 2008] V. S. Sheng, F. Provost, and P. Ipeirotis. Got another label? improving data quality and data mining using multiple, noisy labelers. In *KDD'08*, 2008.

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

[Vits and Gelders, 2002] J. Vits and L. Gelders. Performance improvement theory. *International Journal of Production Economics*, 77(3):285–298, 2002.

[Warby et al., 2014] S. C. Warby, S. L. Wendt, P. Welinder, E. G. S. Munk, O. Carrillo, H. B. D. Sorensen, P. Jennum, P. E. Peppard, P. Perona, and E. Mignot. Sleep-spindle detection: Crowdsourcing and evaluating performance of experts, non-experts and automated methods. *Natural Methods*, 11(4):385–392, 2014.

[Welinder et al., 2010] P. Welinder, S. Branson, P. Perona, and S. J. Belongie. The multidimensional wisdom of crowds. In *NIPS'10*, pages 2424–2432, 2010.

[Zhao et al., 2013] Z. Zhao, D. Yan, W. Ng, and S. Gao. A transfer learning based framework of crowd-selection on twitter. In *KDD'13*, pages 1514–1517, 2013.