Inference Aided Reinforcement Learning for Incentive Mechanism Design in Crowdsourcing

Anonymous Authors[1]

Abstract

Incentive mechanisms are designed to incentivize self-interested workers (e.g. recruited from crowdsourcing) to report high-quality labels. However, existing mechanisms are often developed as one-shot static solutions, assuming that all workers follow the utility-maximizing strategy. In this paper, we build a sequential data acquisition mechanism by firstly developing a Bayesian inference algorithm to estimate workers' labeling strategies from the collected labels. Then, we propose a reinforcement learning algorithm, relying on the above estimates, to uncover how workers respond to different levels of offered payments. Our mechanism determines the payment for workers based on estimated workers' current strategies and the output of the reinforcement learning algorithm. We theoretically prove that our mechanism is able to incentivize workers to provide high-quality labels at equilibrium. We empirically show that our Bayesian inference algorithm can improve the robustness and lower the variance of payment, and our reinforcement learning algorithm performs consistently well on different worker models.

1. Introduction

The ability to quickly collect large scale and high quality labeled datasets is crucial for Machine Learning (ML), and more generally for Artificial Intelligence. Among all proposed solutions, one of the most promising ones is crowdsourcing (Howe, 2006; Slivkins & Vaughan, 2014; Difallah et al., 2015; Simpson et al., 2015). The idea is neat - instead of using a centralized amount of efforts, the to-be-labeled tasks are disseminated to a decentralized crowd of workers to parallelize the collection procedure, leveraging the power of human computation. Nonetheless, it has been noted that crowdsourced data often suffers from quality issues, due to its salient feature of no monitoring and no ground-truth verification of workers' contributed data.

This quality control challenge has been attempted by two relatively disconnected research communities. From the more ML side, quite a few inference techniques have been developed for inferring true labels from crowdsourced and potentially noisy labels (Raykar et al., 2010; Liu et al., 2012; Zhou et al., 2014; Zheng et al., 2017). These solutions often work as one-shot, post-processing procedures facing a static set of workers, whose labeling accuracy is fixed and *informative*. Despite their empirical success, the above methods ignored the effects of *incentives* when dealing with human inputs. It has been observed both in theory and practice that, without appropriate incentive, selfish and rational workers can easily choose to contribute low quality, uninformative, or even malicious data (Sheng et al., 2008; Liu & Chen, 2017b). Existing inference algorithms are very vulnerable in these cases - either much more redundant labels will be needed (low quality inputs), or the methods will simply fail to work (the case with uninformative and malicious inputs).

From the less ML side, the above quality control question has been studied in the context of *incentive mechanism design*. In particular, a family of mechanisms, jointly called *peer prediction*, has been proposed towards addressing the above incentive challenges (Prelec, 2004; Jurca et al., 2009; Witkowski & Parkes, 2012; Dasgupta & Ghosh, 2013). Existing peer prediction mechanisms focus on achieving incentive compatibility (IC), which is defined as truthfully reporting private data, or reporting high quality data, will maximize workers' expected utilities. These mechanisms achieve IC via comparing the reports between the targeted worker, and a randomly selected reference worker, to bypass the challenge of no ground-truth verification. Nonetheless, we note several undesirable properties of existing peer prediction mechanisms. Firstly, from the label inference studies (Zheng et al., 2017), we know that the collected labels contain a wealth of information about the true labels and workers' labeling accuracy. Nonetheless, existing peer prediction mechanisms often rely on the labels of a small set of reference workers, which only represents a limited share of the overall collected information. Secondly, existing peer prediction mechanisms simplify workers' responses to the incentive mechanism by assuming that workers are all fully rational and only follow the utility-maximizing strategy. However, there are evidences showing that human agents may follow bounded-rationality model, and may improve their responding strategies in practice (Simon, 1982; Chas-

tain et al., 2014; Gao et al., 2014). Lastly, it is often assumed that workers' costs in exerting effort to produce high quality labels are known by the mechanism designer.

In this paper, we propose a *learning-based incentive mechanism*, aiming to merge and extend the techniques in the two disconnected areas to address the caveats when they are employed alone, as discussed above. The high level idea is as follows: we divide the large to-be-collected dataset into relatively small task packages. At each step, we employ workers to handle one task package and estimate the true labels and workers' strategies from their reports. Relying on the above estimates, a reinforcement learning (RL) algorithm is used to uncover how workers respond to different levels of offered payments. We determine the payments for workers based on workers' current strategies and the output of the RL algorithm. By doing so, our mechanism not only incentivizes rational workers to provide high-quality labels but also dynamically adjusts the payments according to workers' types.

We summarize our core contributions as follows:

- We propose an incentive compatible RL framework to (i) incentivize high quality labels from workers and (ii) learn to maximize data requester's utility dynamically, without assuming any agent model for workers.
- To achieve incentive compatibility and to calibrate the estimates of label accuracy for training our RL algorithm, we develop a novel Bayesian inference algorithm and theoretically prove its convergence. Since the inference results are approximately corrupted with Gaussian noise, we develop a RL algorithm based on the data-driven Gaussian process regression.
- We provide a novel method to prove the long-term incentive compatibility of RL algorithms.

Besides, we conduct extensive experiments of our algorithms, and the results show that our Bayesian inference algorithm can improve the robustness and lower the variance of payments, which is practically favorable. Meanwhile, our RL algorithm can significantly increase the utility of the data requester under different worker models, such as fully rational and learning agent models.

2. Related Work

Our work is inspired by the following three literatures:

Peer Prediction: This line of works, addressing the incentive issues for reporting high quality data without verification, starts roughly with the seminal works (Prelec, 2004; Gneiting & Raftery, 2007). A series of follow-up works have relaxed various assumptions that have been made(Jurca et al., 2009; Witkowski & Parkes, 2012; Radanovic & Faltings, 2013; Dasgupta & Ghosh, 2013).

Inference method: Recently, inference methods have been applied to crowdsourcing settings, aiming to uncover the true labels from multiple noisy reported copies. Notable successes include EM method (Dawid & Skene, 1979; Raykar et al., 2010; Zhang et al., 2014), Variational Inference (Liu et al., 2012; Chen et al., 2015) and Minimax Entropy Inference (Zhou et al., 2012; 2014). Besides, Zheng et al. (2017) provide a good survey of the existing inference algorithms.

Reinforcement Learning: In the past few years, RL has made several breakthroughs of achieving human-level performance in challenging domains (Mnih et al., 2015; van Hasselt et al., 2016; Silver et al., 2017). More gladly, many studies successfully deploy it to solve some societal problems (Yu et al., 2013; Leibo et al., 2017). Besides, RL has also helped make advances in human-agent collaboration (Sadhu et al., 2016; Wang & Zhang, 2017).

Our work differs from the above literature in the connection between incentive mechanisms and ML. There have been a very few recent studies that share similar taste with ours. For example, to improve the utility of the data requester in crowdsourcing, a multi-armed bandit algorithm is developed in Liu & Chen (2017b) to adjust the state-of-the-art peer prediction mechanism DG13 (Dasgupta & Ghosh, 2013) to a prior-free setting. However, both above bandit algorithm and DG13 still need to assume that workers are fully rational. Instead of randomly choosing a reference worker as commonly done in peer prediction, Liu & Chen (2017a) propose to use supervised learning algorithms to generate the reference reports based on the contextual information of tasks and derive the corresponding IC conditions. In this paper, without assuming the contextual information about tasks, we use Bayesian inference to learn workers' states and true labels, which is an unsupervised-learning algorithm.

3. Problem Formulation

This paper considers the following data collection problem via crowdsourcing: at each discrete time step $t = 1, 2, ...,$ one data requester assigns M tasks with binary answer space $\{1, 2\}$ to $N \geq 3$ candidate workers to label. Workers receive payments for submitting a label for each task. We use $L_i^t(j)$ to denote the label worker i generates for task j at time t. For simplicity of computation, we reserve $L_i^t(j) = 0$ if j is not assigned to i. Furthermore, we use $\mathcal{L}$ and $\boldsymbol{L}$ to denote the set of ground-truth labels and the set of all collected labels respectively.

The generated label $L_i^t(j)$ depends both on the ground-truth $\mathcal{L}(j)$ and worker i's strategy, which is mainly determined by two factors, exerted effort level (high or low) and reporting strategy (truthful or deceitful). Accommodating the notation commonly used in reinforcement learning, we also refer worker i's strategy as his/her internal state. At any given

time for any task, workers at their will adopt an arbitrary combination of effort level and report strategy. Specifically we define $\mathsf{eft}_i^t \in [0,1]$ and $\mathsf{rpt}_i^t \in [0,1]$ as worker i's probability of exerting high efforts and reporting truthfully for task j, respectively. Furthermore, following existing literature (Dasgupta & Ghosh, 2013; Liu & Chen, 2017b), we assume that tasks are homogeneous and workers share the same probability of generating the correct labels if they exert the same level of efforts - we denote these probabilities as $\mathbb{P}_H$ and $\mathbb{P}_L$.[1] We assume $\mathbb{P}_H > \mathbb{P}_L = 0.5$. We further assume that the cost for any worker i to exert low effort is $c_L = 0$, whereas exerting high effort incurs $c_H \geq 0$. [2] These cost parameters stay unknown to the data requester. Worker i's probability of being correct (PoBC) at time t for any given task is then given by

$$\begin{aligned}\mathbb{P}_i^t = \ & \mathsf{rpt}_i^t \cdot \mathsf{eft}_i^t \mathbb{P}_H + (1 - \mathsf{rpt}_i^t) \cdot \mathsf{eft}_i^t (1 - \mathbb{P}_H) + \\ & \mathsf{rpt}_i^t \cdot (1 - \mathsf{eft}_i^t) \mathbb{P}_L + (1 - \mathsf{rpt}_i^t) \cdot (1 - \mathsf{eft}_i^t)(1 - \mathbb{P}_L)\end{aligned} \tag{1}$$

Suppose we assign $m_i^t \leq M$ tasks to worker i at time step t, then his or her utility would be

$$u_i^t = \sum\nolimits_{j=1}^{M} P_i^t(j) - m_i^t \cdot c_H \cdot \mathsf{eft}_i^t \tag{2}$$

where P_i^t denotes our payment to worker i for task j at time t (see Section 4.1 for more details).

At the beginning of each step, the data requester and workers mutually agree to a certain rule of payment determination, which would not be changed until the next time step. The workers are self-interested and may change their strategies according to the expected utility $\mathbb{E}u_i^t$ he/she can get. It is not surprising that workers' different strategies would lead to different PoBCs and then different qualities of labels. After collecting the generated labels, the data requester infers the true labels $\tilde{L}^t(j)$ by running a certain inference algorithm. The aggregate label accuracy A^t and the data requester's utility r_t are defined as follows:

$$\begin{aligned} A^t &= \frac{1}{M} \sum\nolimits_{j=1}^{M} 1\left[\tilde{L}^t(j) = \mathcal{L}(j)\right] \\ r_t &= F(A^t) - \eta \sum\nolimits_{i=1}^{N} \sum\nolimits_{j=1}^{M} P_i^t(j) \end{aligned} \tag{3}$$

where $F(\cdot)$ is a non-decreasing monotonic function mapping accuracy to utility and $\eta > 0$ is a tunable parameter balancing label quality and costs. Naturally, $F(\cdot)$ function is non-deceasing as higher accuracy is preferred.[3]

[1] For simplicity we have assumed that the labeling accuracy is ground-truth label independent.

[2] We make such assumption for simplicity. Our analysis can be extended to the case where both $c_L, c_H \geq 0$, as long as $c_H \geq c_L$.

[3] We use r to denote the data requester's utility as it is used as the reward in our RL algorithm. See Section 4.3 for details.

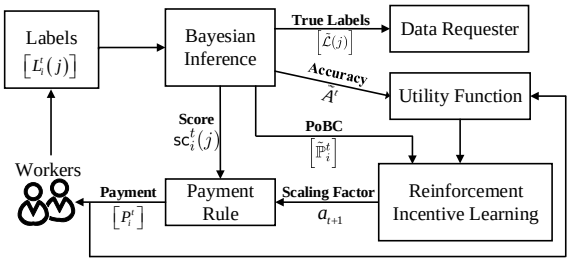


Figure 1. Overview of our incentive mechanism.

4. Incentive Mechanism for Crowdsourcing

Our mechanism mainly consists of three components: one-step payment rule, Bayesian inference and reinforcement incentive learning (RIL); see Figure 1 for the overall layout, where estimated values are denoted with tildes.

The payment rule ensures that reporting truthfully and exerting high efforts is the payment-maximizing strategy for all workers at any given time. Besides, our incentive mechanism as a whole guarantees that this is also the payment-maximizing strategy for all workers in the long-run. We kindly refer readers to Section 5 for the theoretical proof. This property prevents strategic manipulations from workers, which brings more long-term benefits to them by sacrificing short-term gains, or the other way around. The Bayesian inference algorithm is responsible for estimating the true labels, workers' PoBCs and the aggregate label accuracy from the collected labels at each time step. It utilizes soft Dirichlet priors and Gibbs sampling to prevent overestimation of accuracy when workers generate poor-quality labels. RIL adjusts the payment rule based on the historical data of payments, workers' PoBCs and aggregate labels' accuracy, aiming to optimally balance the utility gain from high accuracy and loss from large payments, which corresponds to $F(A^t)$ and $\sum_i \sum_j P_i^t(j)$ in Eqn. (3) respectively.

4.1. Payment Rule

To ensure IC (described in Section 5), we design the payment to worker i for his/her label on task j as

$$P_i^t(j) = a_t \cdot [\mathsf{sc}_i^t(j) - 0.5] + b \tag{4}$$

where $\mathsf{sc}_i^t(j)$ denotes worker i's score on task j, which will be computed by our Bayesian inference algorithm (details in next subsection). $b \geq 0$ is a constant representing the fixed base payment even if the worker purely genrates random labels. We use $a_t \in \mathcal{A}$ to denote the scaling factor, determined by RIL at the beginning of every step t. We assume $\mathcal{A}$ is a finite set $|\mathcal{A}| < \infty$.

4.2. Bayesian Inference

For the simplicity of notations, we omit the superscript t in this subsection. The motivation for designing our own

Algorithm 1 Gibbs Sampling aided Bayesian Inference

1: **Input:** the collected labels $\boldsymbol{L}$, the number of samples W
2: **Output:** the sample sequence $\mathcal{S}$
3: $\mathcal{S} \leftarrow \emptyset$, Initialize $\tilde{\mathcal{L}}$ with the uniform distribution
4: **for** $s = 1$ **to** W **do**
5: **for** $j = 1$ **to** M **do**
6: Compute $\mathbb{P}[\mathcal{L}(j) = k]$ by letting $\mathcal{L}(-j) = \tilde{\mathcal{L}}(-j)$.
7: $\tilde{\mathcal{L}}(j) \leftarrow$ Sample $\{1, 2\}$ with $\mathbb{P}[\mathcal{L}(j) = k]$
8: **end for**
9: Append $\tilde{\mathcal{L}}$ to the sample sequence $\mathcal{S}$
10: **end for**

Bayesian inference algorithm is as follows: we have ran several preliminary experiments using popular inference algorithms used in the literature. Our empirical studies reveal that those methods tend to heavily bias towards overestimating the accuracy when the quality of labels is very low. For example, when there are 10 workers and $\mathbb{P}_i^t = 0.55$, the estimated label accuracy of the EM estimator (Raykar et al., 2010) stays at around 0.9 while the real accuracy is only around 0.5.[4] This heavy bias will lead the data requester's utility r_t to be miscalculated, causing two bad consequences. First, it induces bad incentive property, as workers with poor labeling accuracy now enjoy good estimates. Secondly, this potentially misleads RIL, as r_t is used as reward.

To reduce the inference bias, we develop a Bayesian inference algorithm by introducing soft Dirichlet priors to both the distribution of true labels $\boldsymbol{\tau} = [\tau_1, \tau_2] \sim \text{Dir}(\beta_1, \beta_2)$, where τ_1 and τ_2 denote that of label 1 and 2, respectively, and workers' PoBCs $[\mathbb{P}_i, 1 - \mathbb{P}_i] \sim \text{Dir}(\alpha_1, \alpha_2)$. After doing so, we derive the conditional distribution of true labels given collected labels as (see Appendix I in Supplementary)

$$\mathbb{P}(\mathcal{L}|\boldsymbol{L}) = \mathbb{P}(\boldsymbol{L}, \mathcal{L})/\mathbb{P}(\boldsymbol{L}) \propto B(\hat{\boldsymbol{\beta}}) \prod\nolimits_{i=1}^{N} B(\hat{\boldsymbol{\alpha}}_i) \quad (5)$$

where $B(x, y) = (x-1)!(y-1)!/(x+y-1)!$ denotes the beta function, $\hat{\boldsymbol{\alpha}} = [\hat{\alpha}_1, \hat{\alpha}_2]$, $\hat{\boldsymbol{\beta}} = [\hat{\beta}_1, \hat{\beta}_2]$ and

$$\hat{\alpha}_{i1} = \sum\nolimits_{j=1}^{M} \sum\nolimits_{k=1}^{K} \delta_{ijk} \xi_{jk} + 2\alpha_1 - 1$$
$$\hat{\alpha}_{i2} = \sum\nolimits_{j=1}^{M} \sum\nolimits_{k=1}^{K} \delta_{ij(3-k)} \xi_{jk} + 2\alpha_2 - 1$$
$$\hat{\beta}_k = \sum\nolimits_{j=1}^{M} \xi_{jk} + 2\beta_k - 1$$

where $\delta_{ijk} = \mathbb{1}(L_i(j) = k)$ and $\xi_{jk} = \mathbb{1}(\mathcal{L}(j) = k)$. The convergence of our inference algorithm requires $\alpha_1 > \alpha_2$. To simplify the theoretical analysis in Section 5, we set $\alpha_1 = 1.5$ and $\alpha_2 = 1$ in the rest of this paper.

Note that it is generally hard to derive an explicit formula for the posterior distribution of a specific task j's ground-truth from the conditional distribution $\mathbb{P}(\mathcal{L}|\boldsymbol{L})$. We thus resort to Gibbs sampling for the inference. More specifically, according to Bayes' theorem, we know that the conditional distribution of task j's ground-truth $\mathcal{L}(j)$ satisfies $\mathbb{P}[\mathcal{L}(j)|\boldsymbol{L}, \mathcal{L}(-j)] \propto \mathbb{P}(\mathcal{L}|\boldsymbol{L})$, where $-j$ denotes all tasks excluding j. Leveraging this, we generate samples of the true label vector $\mathcal{L}$ following Algorithm 1. At each step of the sampling procedure (line 6-7), Algorithm 1 first computes $\mathbb{P}[\mathcal{L}(j)|\boldsymbol{L}, \mathcal{L}(-j)]$ and then generates a new sample of $\mathcal{L}(j)$ to replace the old one in $\tilde{\mathcal{L}}$. After traversing through all tasks, Algorithm 1 generates a new sample of the true label vector $\mathcal{L}$. Repeating this process for W times, we get W samples, which is recorded in $\mathcal{S}$. Here, we write the s-th sample as $\tilde{\mathcal{L}}^{(s)}$. Since Gibbs sampling requires a burn-in process, we discard the first W_0 samples in $\mathcal{S}$. After doing so, we calcualte worker i's score on task j as

$$\mathsf{sc}_i^t(j) = \sum\nolimits_{s=W_0+1}^{W} \mathbb{1}(\tilde{\mathcal{L}}^{(s)}(j) = L_i(j))/(W - W_0) \quad (6)$$

and estimate worker i's PoBC $\mathbb{P}_i$ as

$$\tilde{\mathbb{P}}_i = \frac{\sum\limits_{s=W_0+1}^{W} \left[2\alpha_1 - 1 + \sum\limits_{j=1}^{M} \mathbb{1}(\tilde{\mathcal{L}}^{(s)}(j) = L_i(j)) \right]}{(W - W_0) \cdot (2\alpha_1 + 2\alpha_2 - 2 + m_i)} \quad (7)$$

and the distribution of true labels $\boldsymbol{\tau}$ as

$$\tilde{\tau}_k = \frac{\sum\limits_{s=W_0+1}^{W} \left[2\beta_1 - 1 + \sum\limits_{j=1}^{M} \mathbb{1}(\tilde{\mathcal{L}}^{(s)}(j) = k) \right]}{(W - W_0) \cdot (2\beta_1 + 2\beta_2 - 2 + M)}. \quad (8)$$

Furthermore, we define the log-ratio of task j as

$$\tilde{\sigma}_j = \log \frac{\mathbb{P}[\mathcal{L}(j) = 1]}{\mathbb{P}[\mathcal{L}(j) = 2]} = \log \left(\frac{\tilde{\tau}_1}{\tilde{\tau}_2} \prod_{i=1}^{N} \tilde{\lambda}_i^{\delta_{ij1} - \delta_{ij2}} \right) \quad (9)$$

where $\tilde{\lambda}_i = \tilde{\mathbb{P}}_i/(1 - \tilde{\mathbb{P}}_i)$. Finally, we decide the true label estimate $\tilde{\mathcal{L}}(j)$ as 1 if $\tilde{\sigma}_j > 0$ and as 2 if $\tilde{\sigma}_j < 0$. Correspondingly, the label accuracy A is estimated as

$$\tilde{A} = \mathbb{E}(A) = \frac{1}{M} \sum\nolimits_{j=1}^{M} e^{|\tilde{\sigma}_j|} \left(1 + e^{|\tilde{\sigma}_j|} \right)^{-1}. \quad (10)$$

For good inference accuracy, we require both W and W_0 to be large values, and in the rest of this paper, we set $W = 1000$ and $W_0 = 100$ respectively.

4.3. Reinforcement Incentive Learning

In this subsection, we formally introduce our reinforcement incentive learning (RIL) algorithm, which adjusts the scaling factor a_t at each time step t. Viewed under the big picture, it serves as the glue to connect the other components in our mechanism (see the edges and parameters around RIL in Figure 1). To fully understand the technical background, we require the readers to at least be familiar with Q-value and function approximation. For readers with

[4] See Appendix H in Supplementary material for details

limited knowledge, we kindly refer them to Appendix A, where we provide a tutorial on these concepts.

With some transformation, the crowdsourcing problem we aim to tackle in this paper can perfectly fit into the commonly used RL formalization (i.e. a Markov Decision Process). To be more specific, the data requester is the agent and it interacts with workers (i.e. the environment); scaling factors are actions; the utility of the data requester r_t after paying workers (see Eqn. (3)) is the reward; how workers respond to different incentives and potentially change their internal states thereafter forms the transition kernel, which is unobservable; what scaling factor to be picked at each step t given workers' labeling constructs the policy, which needs to be learned. Since the real accuracy cannot be observed, we use the estimated accuracy $\tilde{A}$ calculated by Eqn. (10) instead to construct the reward

$$r_t \approx F(\tilde{A}^t) - \eta \sum\nolimits_{i=1}^{N} P_i^t. \tag{11}$$

To achieve better generalization across different states, it is a common approach to learn a feature-based state representation $\phi(s)$ (Mnih et al., 2015; Liang et al., 2016). Recall that the data requester's implicit utility at time t only depends on the aggregate PoBC averaged across the whole worker body. Such observation already points out to a representation design with good generalization, namely $\phi(s_t) = \sum_{i=1}^{N} \mathbb{P}_i^t / N$. Further recall that, when deciding the current scaling factor a_t, the data requester does not observe the latest workers' PoBCs and thus cannot directly estimate the current $\phi(s_t)$. Due to this one-step delay, we have to build our state representation using the previous observation. Since most workers would only change their internal states after receiving a new incentive, there exists some imperfect mapping function $\phi(s_t) \approx f(\phi(s_{t-1}), a_{t-1})$. Utilizing this implicit function, we introduce the augmented state representation in RIL as

$$\hat{s}_t = \langle \phi(\hat{s}_{t-1}), a_{t-1} \rangle.$$

Since neither r_t nor $\hat{s}_t$ can be perfectly accurate, it would not be a surprise to observe some noise that cannot be directly learned in our Q-function. As for most crowdsourcing problems the number of tasks M is large, we leverage the central limit theorem to justify our modeling of the noise using a Gaussian process. To be more specific, we calculate the temporal difference (TD) error as

$$r_t \approx Q^\pi(\hat{s}_t, a_t) - \gamma \mathbb{E}_\pi Q^\pi(\hat{s}_{t+1}, a_{t+1}) + \epsilon_t \tag{12}$$

where the noise ϵ_t follows a Gaussian process $\mathcal{N}(\hat{s}_t, \hat{s}_{t+1})$, and $\pi = \mathbb{P}(a|\hat{s})$ denotes the current policy. Doing so, we gain two benefits. First, it greatly simplifies the derivation of the update equation for the Q-function. Secondly, as shown in our empirical results later, it is robust against different worker models. Besides, following Gasic & Young (2014)

Algorithm 2 Reinforcement Incentive Learning (RIL)

1: **for** each episode **do**
2: **for** each step in the episode **do**
3: Decide the scaling factor as (ϵ-greedy method)

$$a_t = \begin{cases} \arg\max_{a \in \mathcal{A}} Q(\hat{s}_t, a) & \text{Probability } 1-\epsilon \\ \text{Random } a \in \mathcal{A} & \text{Probability } \epsilon \end{cases}$$

4: Assign tasks and collect labels from the workers
5: Run Bayesian inference to get $\hat{s}_{t+1}$ and r_t
6: Use $(\hat{s}_t, a_t, r_t)$ to update $\boldsymbol{K}$, $\boldsymbol{H}$ and $\boldsymbol{r}$ in Eqn. (14)
7: **end for**
8: **end for**

we approximate Q-function as

$$Q^\pi(\hat{s}_{t+1}, a_{t+1}) \approx \mathbb{E}_\pi Q^\pi(\hat{s}_{t+1}, a_{t+1}) + \epsilon_\pi$$

where ϵ_π also follows a Gaussian process.

Under the Gaussian process approximation, all the observed rewards and the corresponding Q values up to the current step t form a equation set, and it can be written as

$$\boldsymbol{r} = \boldsymbol{H}\boldsymbol{Q} + \boldsymbol{N} \tag{13}$$

where $\boldsymbol{r}$, $\boldsymbol{Q}$ and $\boldsymbol{N}$ denote the collection of rewards, Q values, and residuals. Following Gaussian process's assumption for residuals, $\boldsymbol{N} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\sigma}^2)$, where $\boldsymbol{\sigma}^2 = \text{diag}(\sigma^2, \ldots, \sigma^2)$. The matrix $\boldsymbol{H}$ satisfies $\boldsymbol{H}(k,k) = 1$ and $\boldsymbol{H}(k, k+1) = -\gamma$ for $k = 1, \ldots, t$. Then, by using the online Gaussian process regression algorithm (Engel et al., 2005), we effectively learn the Q-function as

$$Q(\hat{s}, a) = \boldsymbol{k}(\hat{s}, a)^{\mathrm{T}} (\boldsymbol{K} + \boldsymbol{\sigma}^2)^{-1} \boldsymbol{H}^{-1} \boldsymbol{r} \tag{14}$$

where $\boldsymbol{k}(\hat{s}, a) = [k((\hat{s}, a), (\hat{s}_1, a_1)), \ldots, k((\hat{s}, a), (s_t, a_t))]^{\mathrm{T}}$ and $\boldsymbol{K} = [\boldsymbol{k}(\hat{s}_1, a_1), \ldots, \boldsymbol{k}(\hat{s}_t, a_t)]$. Here, we use $k(\cdot, \cdot)$ to denote the Gaussian kernel. Finally, we employ the classic ϵ-greedy method to decide a_t based on the learned Q-function. To summarize, we provide a formal description about our RL algorithm in Algorithm 2. Note that, when updating $\boldsymbol{K}$, $\boldsymbol{H}$ and $\boldsymbol{r}$ in Line 6, we employ the sparse approximation to discard some data so that the size of these matrices will not increase infinitely. The details of this technique can be found in Gasic & Young (2014).

5. Convergence and Incentive Results

In this section, we present the theoretic analysis for our incentive mechanism[5]. Our main results are as follows:

Theorem 1 (One-Step IC)**.** *In any time step t, when $M \gg 1$, $(2\mathbb{P}_H)^{2(N-1)} \geq M$ and $a_t > \frac{c_H}{\mathbb{P}_H - 0.5}$, reporting truthfully and exerting high efforts is the utility-maximizing strategy for any worker i if the other workers all follow this strategy.*

[5] Currently, our theoretical analysis is for the case that $m_i^t = M$. $m_i^t < M$ requires to replace the binomial distribution with the trinomial distribution when analyzing a key function involved in the proof in the supplementary file. The main idea of proof will be the same, and we will extend our proof in the future work.

Theorem 2 (Long-Term IC)**.** *Suppose the conditions in Proposition 1 are satisfied and the learned Q-function approaches the real $Q^\pi(\hat{s}, a)$. When*

$$\eta M(N-1)\mathbb{P}_H \cdot G_{\mathcal{A}} > \frac{F(1) - F(1-\psi)}{1-\gamma} \tag{15}$$

$$\psi = (\tau_1\tau_2^{-1} + \tau_1^{-1}\tau_2)[4\mathbb{P}_H(1-\mathbb{P}_H)]^{\frac{N-1}{2}} \tag{16}$$

always reporting truthfully and exerting high efforts is the utility-maximizing strategy for any worker i in the long-term if the other workers all follow this strategy. Here, $G_{\mathcal{A}} = \min_{a,b\in\mathcal{A},a\neq b}|a-b|$ denotes the minimal gap between two available values of the scaling factor.

In the following two subsections, we will provide the outline of the proofs of these two theorems. Note that, in addition to incentive compatiblity, our mechanism can also get rid of the undesired uninformative equlibrium that most peer prediction mechanisms have. See Appendix G in Supplementary for detailed discussion. Besides, in our proof, if we omit the superscript t in an equation, we mean that this equation holds for all time steps. Besides, we employ the convention that $\bar{\mathbb{P}} = 1 - \mathbb{P}$, $\hat{\mathbb{P}} = \max\{\mathbb{P}, \bar{\mathbb{P}}\}$ and $\mathbb{P}_0 = \tau_1$.

5.1. Proof for One-Step IC

When $M \gg 1$ and $a_t > c_H/(\mathbb{P}_H - 0.5)$, if $\tilde{\mathbb{P}}_i^t \approx \mathbb{P}_i^t$, worker i's utility-maximizing strategy will be reporting truthfully and exerting high efforts.[6] Thus, we can conclude Theorem 1 by proving the convergence of our Bayesian inference algorithm, i.e. proving $\tilde{\mathbb{P}}_i^t \approx \mathbb{P}_i^t$. $\tilde{\mathbb{P}}_i^t$ is computed according to Eqn. (7), and if most of the samples in our Bayesian inference are correct, namely $\tilde{\mathcal{L}}^{(s)}(j) \equiv \mathcal{L}(j)$, we can prove $\tilde{\mathbb{P}}_i^t \approx \mathbb{P}_i^t$ by leveraging the law of large numbers. This observation motivates us to bound $|\tilde{\mathbb{P}}_i^t - \mathbb{P}_i^t|$ by calculating the upper bound of the ratio of wrong labels in the samples. Thereby, we prove Theorem 1 with the following two steps:

Step 1: In this step, we aim to derive the upper bound of the ratio of wrong samples. To achieve this objective , we introduce n and m to denote the number of tasks of which the true label sample in Eqn. (7) is correct ($\tilde{\mathcal{L}}^{(s)}(j) = \mathcal{L}(j)$) and wrong ($\tilde{\mathcal{L}}^{(s)}(j) \neq \mathcal{L}(j)$), respectively. Then, we are able to prove the following lemma:

Lemma 1. *When $M \gg 1$,*

$$\mathbb{E}[m/M] \lesssim (1+e^\delta)^{-1}(\varepsilon + e^\delta)(1+\varepsilon)^{M-1} \tag{17}$$

$$\mathbb{E}[m/M]^2 \lesssim (1+e^\delta)^{-1}(\varepsilon^2 + e^\delta)(1+\varepsilon)^{M-2} \tag{18}$$

where $\varepsilon^{-1} = \prod_{i=0}^{N}(2\hat{\mathbb{P}}_i)^2$, $\delta = O[\Delta \cdot \log(M)]$ and

$$\Delta = \sum\nolimits_{i=1}^{N}[1(\mathbb{P}_i < 0.5) - 1(\mathbb{P}_i > 0.5)].$$

We defer the detailed proof to Appendix C in Supplementary. Our main idea is to introduce a set of counts for the collected labels at first. More specifically, among the n tasks of which the posterior true label is correct, x_i and y_i denote the number of tasks of which worker i's label is correct and wrong, respectively. Among the remaining m tasks, w_i and z_i denote the number of tasks of which worker i's label is correct and wrong, respectively. Then, we calculate the approximation of $\mathbb{P}(m)$ based on the conditional probabilities $\mathbb{P}(x_i, y_i, w_i, z_i|m)$ and $\mathbb{P}(\mathcal{L}|\boldsymbol{L})$. The upper bounds of $\mathbb{E}[m/M]$ and $\mathbb{E}[m/M]^2$ can be obtained by calculating the upper bounds of $\sum_m m\mathbb{P}(m)$ and $\sum_m m^2\mathbb{P}(m)$.

Step 2: In this step, we derive the upper bound of $|\tilde{\mathbb{P}}_i - \mathbb{P}_i|$ under the conditions of Theorem 1. Following the notations in Step 1, when $M \gg 1$, in Eqn. (7), we can have $\tilde{\mathbb{P}}_i \approx \mathbb{E}_{\mathcal{L}}(x_i + z_i)/M$, where $\mathbb{E}_{\mathcal{L}}$ denotes the expectation against $\mathbb{P}(\mathcal{L}|\boldsymbol{L})$. Meanwhile, according to the law of large numbers, $\mathbb{P}_i \approx (x_i + w_i)/M$. Thus, we can have

$$|\tilde{\mathbb{P}}_i - \mathbb{P}_i| \approx \mathbb{E}_{\mathcal{L}}|w_i - z_i|/M \leq \mathbb{E}_{\mathcal{L}}\left[m/M\right]. \tag{19}$$

If workers except for worker i all report truthfully and exert high efforts, then $\Delta \leq -1$ in Theorem 1 because we require $N \geq 3$ in Section 3. Thus, $e^\delta \approx 0$. Since $2\hat{\mathbb{P}}_i \geq 1$, we have $\varepsilon^{-1} \geq (2\mathbb{P}_H)^{2(N-1)}$. Hence, $\varepsilon \leq M^{-1}$ when $(2\mathbb{P}_H)^{2(N-1)} \geq M$. Taking the above analysis into consideration, Eqn. (17) and (18) can be calculated as

$$\mathbb{E}\left[\frac{m}{M}\right] \lesssim \frac{C_1}{M \cdot C_2} \ , \ \mathbb{E}\left[\frac{m}{M}\right]^2 \lesssim \frac{C_1}{M^2 \cdot C_2^2} \tag{20}$$

where $C_1 = (1+M^{-1})^M \approx e$ and $C_2 = 1 + M^{-1} \approx 1$. Then, $m/M \approx 0$ because $\mathbb{E}[m/M] \approx 0$ and $\mathrm{Var}[m/M] = \mathbb{E}[m/M]^2 - (\mathbb{E}[m/M])^2 \approx 0$. In this case, $\tilde{\mathbb{P}}_i \approx \mathbb{P}_i$, and thus we can conclude Theorem 1.

5.2. Proof for Long-Term IC

Due to the one-step IC in our mechanism, we know that, to get higher long-term payments, worker i must mislead our RL algorithm into at least increasing the scaling factor from a to any $a' > a$ at a certain state $\hat{s}$. Actually, our RL algorithm will only increase the scaling factor when the state-action value function satisfies $Q^\pi(\hat{s}, a) \leq Q^\pi(\hat{s}, a')$. Eqn. (11) tells us that our objective function consists of the utility obtained from the collected labels ($F(\tilde{A}^t)$) and the utility lost in the payment ($\eta\sum_{i=1}^{N} P_i^t$). Once we increase the scaling factor, we at least need to increase the payments for the other $N-1$ workers by $M(N-1)\mathbb{P}_H \cdot G_{\mathcal{A}}$, which corresponds to the left-hand side of Eqn. (15).

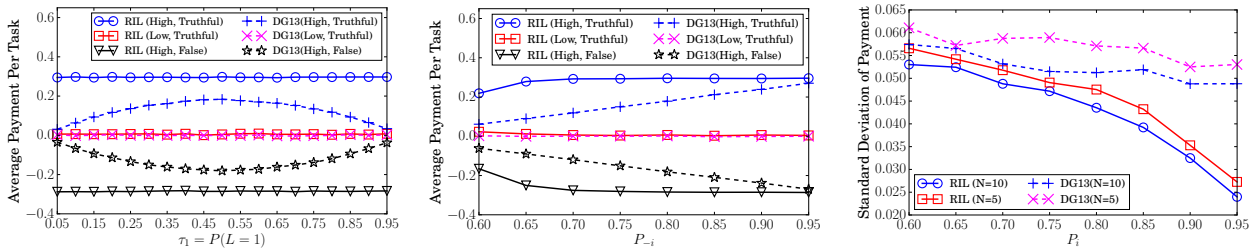
On the other hand, for the obtained utility, we can have

Lemma 2. *In any time step t, if all workers except for worker i report truthfully and exert high efforts,*

$$F(\tilde{A}^t) \leq F(1) \ , \ F(\tilde{A}^t) \geq F(1-\psi)$$

where ψ is defined in Eqn. (16).

[6] See Appendix F in Supplementary for more details.



(a) (b) (c)

Figure 2. Empirical analysis on our Bayesian inference algorithm. (a) Average payment per task given true label's distribution. (b) Average payment per task given PoBCs of workers excluding i. (c) The standard deviation of the payment given worker i's PoBC.

We again defer the detailed proof to Appendix E in Supplementary. Our main idea is to derive the bounds of $\tilde{A}$ by analyzing the distribution of $\tilde{\sigma}_j$. Our motivation comes form Eqn. (10) which ensures, $\tilde{A} \approx 1 - \mathbb{E}[1/(1 + e^{|\tilde{\sigma}_j|})]$. From Lemma 2, we can know that, even in the long-term, worker i can at most increase our value by $(1-\gamma)^{-1}[F(1) - F(1-\psi)]$, which corresponds to the right-hand side of Eqn. (15).

Thereby, if Eqn. (15) is satisfied, worker i will be unable to make up our value loss increment in the payments, and our RL algorithm will reject the hypothesis to increase the scaling factor. In this case, the only utility-maximizing strategy for worker i is to report truthfully and exert high efforts in all time steps, which concludes Theorem 2.

6. Empirical Experiments

In our incentive mechanism, Bayesian inference extracts important and useful information from all collected labels. On the other hand, RIL empowers us with the adaptivity to different types of worker models. In this section, we demonstrate these benefits via empirical studies.

6.1. Performance Analysis of Bayesian Inference

In this subsection, we focus on our Bayesian inference algorithm by fixing the scaling factor $a_t = 1$ and setting $M = 100$, $N = 10$, $\mathbb{P}_H = 0.8$, $b = 0$ and $m_i^t = M(N-1)/N$. We use the average payment to worker i as a proxy, as it is proportional to scores computed by Bayesian inference. We use DG13, the state-of-the-art peer prediction mechanism for binary labels (Dasgupta & Ghosh, 2013), as the benchmark to conduct our comparison. To set up the experiments, we generate task j's true label $\mathcal{L}(j)$ following its distribution $\boldsymbol{\tau}$ (to be specified) and worker i's label for task j based on i's PoBC $\mathbb{P}_i$ and $\mathcal{L}(j)$. For each data point, we run experiments for 1000 rounds and report the mean.

In Figure 2a, we let all workers excluding i report truthfully and exert high efforts (i.e. $\mathbb{P}_{-i} = \mathbb{P}_H$), and increase τ_1 from 0.05 to 0.95. In Figure 2b, we let $\tau_1 = 0.5$, and increase other workers' PoBCs $\mathbb{P}_{-i}$ from 0.6 to 0.95. As both figures reveal, in our mechanism, the payment for worker i almost only depends on his/her own strategy. On contrast, in DG13, the payments severely affected by the distribution of true labels and the strategies of other workers. In other words, our Bayesian inference is more robust to different environments. Furthermore in Figure 2c, we present the standard deviation of the payment to worker i. We let $\tau_1 = 0.5$, $\mathbb{P}_{-i} = \mathbb{P}_H$ and increase $\mathbb{P}_i$ from 0.6 to 0.95. As shown in the figure, our method manages to achieve a noticeably smaller standard deviation compared to DG13. In summary, compared with traditional peer prediction mechanisms that only make use of a small set of workers' labels to decide a payment, the usage of our Bayesian inference algorithm improves its robustness and decreases the variance, because of its ability to fully exploit all the collected labels.

6.2. Empirical Analysis on RIL

In this subsection, we focus on investigating whether RIL proposed in Section 4.3 consistently manages to learn a good policy to maximize the data requester's cumulative utility $R = \sum_t r_t$. For all the experiments in this subsection, we set $M = 100$, $N = 10$, $\mathbb{P}_H = 0.8$, $b = 0$, $c_H = 0.02$, the available value set of the scaling factor $\mathcal{A} = \{0.1, 1.0, 5.0, 10\}$, the exploration rate $\epsilon = 0.2$ for RIL, $F(A) = A^{10}$, $\eta = 0.1$ for the utility function (Eqn. (3)) and the number of time steps for an episode as 28. To reduce the influence of outliers, we report the average over 5 trials. To demonstrate our algorithm's general applicability, we test it under three different worker models, with each capturing a different rationality level. The formal description of the three models is as follows:

- **Rational** workers alway act to maximize their own utilities. Since our incentive mechanism theoretically ensures that exerting high effort is the utility-maximizing strategy for all workers (proved in Section 5), it is safe to assume workers always do so as long as the payment is high enough to cover the cost.
- **Quantal Response (QR)** workers (McKelvey & Palfrey, 1995) exert high efforts with the probability

$$\mathsf{eft}_i^t = \frac{\exp(\lambda \cdot u_{iH}^t)}{\exp(\lambda \cdot u_{iH}^t) + \exp(\lambda \cdot u_{iL}^t)}$$

where u_{iH}^t and u_{iL}^t denote worker i's expected utility after exerting high or low efforts respectively at time t. λ describe workers' rationality level and we set $\lambda = 3$.

- **Multiplicative Weight Update (MWU)** workers (Chastain et al., 2014) update their probabilities of exerting high efforts at every time step t after receiving the payment as the following equation

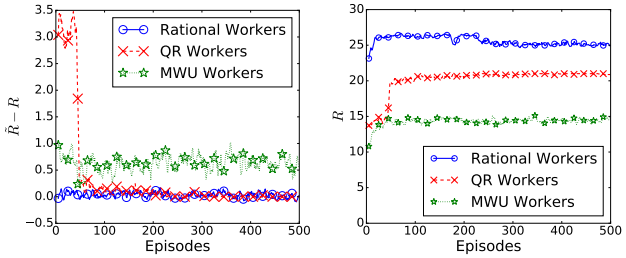
$$\mathsf{eft}_i^{t+1} = \frac{\mathsf{eft}_i^t(1 + \bar{u}_{\cdot H})}{\mathsf{eft}_i^t(\bar{u}_{\cdot H} - \bar{u}_{\cdot L}) + \bar{u}_{\cdot L} + 1}$$

where $\bar{u}_{\cdot H}$ and $\bar{u}_{\cdot L}$ denote the average utilities received if exerting high efforts or low efforts at time t respectively. We initialize eft_i^0 as 0.2 in our experiments.

Since we have shown the advantage of using Bayesian inference in Section 6.1, for the sake of fair comparison, we use our payment rule with manually adjusted scaling factor as the benchmark. In this case, how tasks are assigned does not affect the empirical analysis we wish to conduct and thus we let every worker to be assigned the whole task set at each time step t (i.e. $\forall i, m_i^t = M$).

Our first set of experiments focus on the estimation bias of the the data requester's cumulative utility R. Since the data requester's utility is used as the reward in RIL, it would not be a surprise that the reliability of the estimation of the reward plays a crucial role in determining the well-being of the whole mechanism. As Figure 3a shows, the estimated value only deviates from the real one in a very small magnitude after a few episodes of learning, regardless of which worker model the experiments run on. The next set of experiments is about how quickly RIL learns. As Figure 3b shows, under all three worker models, RIL manages to pick up and stick to a promising policy in less than 100 episodes. This observation also demonstrates the robustness of RIL under different deploying environments.

Lastly, we take the learned policy after 500 episodes with exploration rate turned off (i.e. $\epsilon = 0$) and compares it with two benchmarks constructed by ourself (see Table 1). To create the first one, Fixed Optimal, we try all 4 possible fixed value for the scaling factor and report the highest cumulative reward realized by either of them. Note most traditional peer prediction mechanisms assume a fixed scaling factor and thus Fixed Optimal represents the best performance possibly achieved by them. To create the second one, Adaptive Optimal, we change the scaling factor every 4 steps and report the highest cumulative reward via traversing all $4^7 = 16384$ possible configurations. This benchmark is infeasible to be reproduced in real-world practice, once the number of times steps becomes large. Yet it is very close to the global optimal in the sequential setting. As Table 1 demonstrates, the two benchmarks plus RIL all achieve a similar performance tested for rational and QR workers. This is because these two kinds of workers have a fixed pattern in response



(a) (b)

Figure 3. Empirical analysis on our RL algorithm. (a) The gap between the estimation of the data requester's cumulative rewards and the real one, smoothed over 5 episodes. (b) The learning curve of our mechanism smoothed over 5 episodes.

Table 1. Performance comparison on three worker models. Data requester's cumulative utility normalized over the number of tasks. Standard deviation reported in parenthesis.

METHOD	RATIONAL	QR	MWU
FIXED OPTIMAL	27.584 (.253)	21.004 (.012)	11.723 (.514)
ADAPTIVE OPTIMAL	27.618 (.109)	21.017 (.004)	18.475 (.382)
RIL	27.184 (.336)	21.016 (.018)	15.726 (.416)

to incentives and thus the optimal policy would be a fixed scaling factor through all the whole episode. On contrast, MWU workers' learn utility-maximizing strategies change gradually, and the learning process is affected by the incentives. Under this model, compared with Fixed Optimal, RIL increases the cumulative reward from 11.7 to 15.6, which is a significant improvement considering the unreachable real optimal is only around 18.5. Up to this point, with three sets of experiments, we demonstrate the competitiveness of RIL and its robustness under different environment.

7. Future Work and Conclusion

In this paper, we build a novel RL framework for sequentially acquiring data from crowdsourcing. At each time step, our mechanism uses the Bayesian inference algorithm to learn workers' probability of being correct, and we issue payments to workers that are proportional to these estimated accuracy. When interacting with workers, our mechanism learns the optimal policy to adjust the scaling factor of the payments via our reinforcement incentive learning algorithm. We theoretically prove that our mechanism is incentive compatible. As a by product, we have also proved the convergence of our Bayesian inference method. We empirically show that our Bayesian inference algorithm can help improve the robustness and lower the variance of payments, which are favorable properties in practice. Meanwhile, our reinforcement incentive learning algorithm performs consistently well with different worker models. In the future, for more practical cases where the mapping between workers and tasks is very sparse, we will explore the question that how to improve our mechanism with more complex state representations, for example, using neural networks.

References

Chastain, Erick, Livnat, Adi, Papadimitriou, Christos, and Vazirani, Umesh. Algorithms, games, and evolution. *PNAS*, 111(29):10620–10623, 2014.

Chen, Xi, Lin, Qihang, and Zhou, Dengyong. Statistical decision making for optimal budget allocation in crowd labeling. *Journal of Machine Learning Research*, 16: 1–46, 2015.

Dasgupta, Anirban and Ghosh, Arpita. Crowdsourced judgement elicitation with endogenous proficiency. In *Proc. of WWW*, 2013.

Dawid, Alexander Philip and Skene, Allan M. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics*, pp. 20–28, 1979.

Difallah, Djellel Eddine, Catasta, Michele, Demartini, Gianluca, Ipeirotis, Panagiotis G, and Cudré-Mauroux, Philippe. The dynamics of micro-task crowdsourcing: The case of amazon mturk. In *Proc. of WWW*, 2015.

Engel, Yaakov, Mannor, Shie, and Meir, Ron. Reinforcement learning with gaussian processes. In *Proc. of ICML*, 2005.

Gao, Xi Alice, Mao, Andrew, Chen, Yiling, and Adams, Ryan Prescott. Trick or treat: putting peer prediction to the test. In *Proc. of ACM EC*, 2014.

Gasic, Milica and Young, Steve. Gaussian processes for pomdp-based dialogue manager optimization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(1):28–40, 2014.

Gneiting, Tilmann and Raftery, Adrian E. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

Howe, Jeff. The rise of crowdsourcing. *Wired Magazine*, 14(6), 06 2006. URL `http://www.wired.com/wired/archive/14.06/crowds.html`.

Jurca, Radu, Faltings, Boi, et al. Mechanisms for making crowds truthful. *Journal of Artificial Intelligence Research*, 34(1):209, 2009.

Leibo, Joel Z., Zambaldi, Vinicius, Lanctot, Marc, Marecki, Janusz, and Graepel, Thore. Multi-agent reinforcement learning in sequential social dilemmas. In *Proc. of AAMAS*, 2017.

Liang, Yitao, Machado, Marlos C., Talvitie, Erik, and Bowling, Michael. State of the art control of atari games using shallow reinforcement learning. In *Proc. of AAMAS*, 2016.

Liu, Qiang, Peng, Jian, and Ihler, Alexander T. Variational inference for crowdsourcing. In *Proc. of NIPS*, 2012.

Liu, Yang and Chen, Yiling. Machine-learning aided peer prediction. In *Proc. of ACM EC*, 2017a.

Liu, Yang and Chen, Yiling. Sequential peer prediction: Learning to elicit effort using posted prices. In *AAAI*, pp. 607–613, 2017b.

McKelvey, Richard D and Palfrey, Thomas R. Quantal response equilibria for normal form games. *Games and economic behavior*, 10(1):6–38, 1995.

Mnih, Volodymyr, Kavukcuoglu, Koray, Silver, David, Rusu, Andrei A., Veness, Joel, Bellemare, Marc G., Graves, Alex, Riedmiller, Martin, Fidjeland, Andreas K., Ostrovski, Georg, Petersen, Stig, Beattie, Charles, Sadik, Amir, Antonoglou, Ioannis, King, Helen, Kumaran, Dharshan, Wierstra, Daan, Legg, Shane, and Hassabis, Demis. Human-level Control through Deep Reinforcement Learning. *Nature*, 518(7540):529–533, 02 2015.

Prelec, Dražen. A bayesian truth serum for subjective data. *science*, 306(5695):462–466, 2004.

Radanovic, Goran and Faltings, Boi. A robust bayesian truth serum for non-binary signals. In *Proc. of AAAI*, 2013.

Raykar, Vikas C, Yu, Shipeng, Zhao, Linda H, Valadez, Gerardo Hermosillo, Florin, Charles, Bogoni, Luca, and Moy, Linda. Learning from crowds. *Journal of Machine Learning Research*, 11(Apr):1297–1322, 2010.

Sadhu, Vidyasagar, Salles-Loustau, Gabriel, Pompili, Dario, Zonouz, Saman A., and Sritapan, Vincent. Argus: Smartphone-enabled human cooperation via multi-agent reinforcement learning for disaster situational awareness. *2016 IEEE International Conference on Autonomic Computing (ICAC)*, pp. 251–256, 2016.

Sheng, Victor S, Provost, Foster, and Ipeirotis, Panagiotis G. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proc. of SIGKDD*, 2008.

Silver, David, Schrittwieser, Julian, Simonyan, Karen, Antonoglou, Ioannis, Huang, Aja, Guez, Arthur, Hubert, Thomas, Baker, Lucas, Lai, Matthew, Bolton, Adrian, Chen, Yutian, Lillicrap, Timothy, Hui, Fan, Sifre, Laurent, van den Driessche, George, Graepel, Thore, and Hassabis, Demis. Mastering the game of go without human knowledge. *Nature*, 550:354 EP –, 10 2017.

Simon, Herbert Alexander. *Models of bounded rationality: Empirically grounded economic reason*, volume 3. MIT press, 1982.

Simpson, Edwin D, Venanzi, Matteo, Reece, Steven, Kohli, Pushmeet, Guiver, John, Roberts, Stephen J, and Jennings, Nicholas R. Language understanding in the wild: Combining crowdsourcing and machine learning. In *Proc. of WWW*, 2015.

Slivkins, Aleksandrs and Vaughan, Jennifer Wortman. Online decision making in crowdsourcing markets: Theoretical challenges. *ACM SIGecom Exchanges*, 12(2):4–23, 2014.

van Hasselt, Hado, Guez, Arthur, and Silver, David. Deep reinforcement learning with double q-learning. In *AAAI*, 2016.

Wang, Yue and Zhang, Fumin. *Trends in Control and Decision-Making for Human-Robot Collaboration Systems*. Springer Publishing Company, Incorporated, 1st edition, 2017. ISBN 3319405322, 9783319405322.

Witkowski, Jens and Parkes, David C. Peer prediction without a common prior. In *Proc. of ACM EC*, 2012.

Yu, Chao, Zhang, Minjie, and Ren, Fenghui. Emotional multiagent reinforcement learning in social dilemmas. In *PRIMA*, 2013.

Zhang, Yuchen, Chen, Xi, Zhou, Denny, and Jordan, Michael I. Spectral methods meet em: A provably optimal algorithm for crowdsourcing. In *Proc. of NIPS*, 2014.

Zheng, Yudian, Li, Guoliang, Li, Yuanbing, Shan, Caihua, and Cheng, Reynold. Truth inference in crowdsourcing: is the problem solved? *Proc. of the VLDB Endowment*, 10(5):541–552, 2017.

Zhou, Dengyong, Liu, Qiang, Platt, John, and Meek, Christopher. Aggregating ordinal labels from crowds by minimax conditional entropy. In *Proc. of ICML*, 2014.

Zhou, Denny, Basu, Sumit, Mao, Yi, and Platt, John C. Learning from the wisdom of crowds by minimax entropy. In *Proc. of NIPS*, 2012.