A Reinforcement Learning Framework for Eliciting High Quality Information

Zehong Hu[1,2], Yang Liu[3], Yitao Liang[4] and Jie Zhang[2]
1 Rolls-Royce@NTU Corporate Lab, Nanyang Technological University, Singapore
2 School of Computer Science and Engineering, Nanyang Technological University, Singapore
3 Harvard University, USA
4 University of California, Los Angeles, USA

Abstract

Peer prediction is a class of mechanisms that help elicit high-quality information from strategic human agents when there is no ground-truth for verifying contributions. Despite its elegant design, peer prediction mechanisms often fail in practice, mostly due to two shortcomings: (1) agents' incentives for exerting effort to produce high-quality information are assumed to be known; (2) agents are modeled as being fully rational. In this paper, we propose the first reinforcement learning (RL) framework in this domain, *Reinforcement Peer Prediction*, to tackle these two limitations. In our framework, we develop a RL algorithm for the data requester to dynamically adjust the scaling level to maximize his revenue, and to pay workers using peer prediction scoring functions. Experiments show significant improvement in data requestor's revenue under different agent models.

Introduction

Crowdsourcing rises as a promising inexpensive method to collect a large amount of training data quickly (Snow et al. 2008; Deng et al. 2009). Notwithstanding its high efficiency, one salient concern about crowdsourcing is the quality of the collected information, as it is often too costly to verify workers' contributions. This problem is called information elicitation without verification (Waggoner and Chen 2014). A class of incentive mechanisms, collectively called peer prediction, has been developed to solve this problem (Miller, Resnick, and Zeckhauser 2005; Jurca, Faltings, and others 2009; Witkowski and Parkes 2012b; 2012a; Radanovic and Faltings 2013). The core idea of peer prediction is quite simple and elegant – the mechanism designer financially incentivizes workers according to the scoring of their contributions in comparison with their peers'. The payment rules are designed so that each worker reporting truthfully or reporting a high-quality signal is a strict Bayesian Nash Equilibrium for all workers.

Many peer prediction mechanisms adopt the effort-sensitive model to depict agents' trade-off reasoning in contributing high-quality information (Witkowski et al. 2013; Dasgupta and Ghosh 2013; Shnayder et al. 2016; Liu and Chen 2017). In these mechanisms, workers are incentivized to exert high effort to generate high-quality answers. One critical assumption in those mechanisms is an explicitly-known worker model which includes workers' incentives. Furthermore it is also assumed workers are fully rational, following the utility-maximizing strategy. Unfortunately, neither is true in practice. Firstly, workers' incentives to exert high effort can most likely only be known after we, as the mechanism designers, interact with them. Secondly, there is strong evidence showing that human workers are not fully rational, and they are often observed to be deviating from equilibrium strategies in practice (McKelvey and Palfrey 1995; Jurca and Faltings 2007; Gao et al. 2014). Note (Gao et al. 2014) explicitly pointed this issue out in the peer prediction setting.

To push peer prediction mechanisms towards being more practical, we propose a reinforcement learning (RL) framework, *Reinforcement Peer Prediction*, to interact with workers, so as to (1) incentivize workers to converge to a high effort exertion state, and (2) learn the optimal payment based on workers' contributions at each step. Nevertheless, we face two main challenges. Firstly, classic reinforcement learning focuses on the interaction between a single agent and its environment. We, instead, need to effectively consider a multi-agent setting. Immediately more convoluted state evolution processes are formed among workers, because our incentive strategy relies on the comparison between workers' answers and their peers'. Specifically, the evolution of workers' state is a outcome of collective actions from all workers, as well as the environment. Secondly, no ground-truth is available to evaluate either the state of the workers or the rewards in our setting, whereas it is taken as granted in most other reinforcement learning frameworks. Due to these two challenges, until a proper way to evaluate workers' contributions is proposed, no model-free RL algorithms which learn from reward signals can be applied.

Integrating both fields, reinforcement learning and peer prediction, we could successfully use one's advantage to resolve the other's limitation, and as a whole an approach appealing to a broader interest potentially emerges. More specifically, the main contributions of this paper are as follows. (1) We propose the first reinforcement peer prediction mechanism which does not need to assume a decision-making model for workers and removes uninformative equilibrium where agents report uninformative information. Our mechanism combines a peer prediction mechanism with re-

inforcement learning to jointly incentivize workers and learn the optimal scaling level at each step. (2) Due to the lack of ground-truth, we adopt Bayesian inference to evaluate workers' contributions, and to infer the reward following each action (i.e. offered scaling level). We derive the explicit posterior distribution of workers' contributions and employ Gibbs sampling to eliminate its bias. (3) In our setting, the inferred contributions are corrupted by noise and only the most recent previous state rather than the current can be observed. We use the online Gaussian process regression to learn the Q-function and replace the unknown current state with the pair of the last observed state and scaling level (action). (4) We conduct empirical evaluation, and the results show that our mechanism is robust, and is able to significantly increase data requester's revenue under different worker models, such as fully rational, bounded rational and learning agent models.

Problem Formulation

Our proposed mechanism mainly works in the setting where one data requester, at every step, assigns M binary tasks with answer space $\{1, 2\}$ to $N \geq 4$ candidate workers. At step t, worker i's labels for the task j is denoted as $L_i^t(j)$, and correspondingly the payment that the data requester pays is denoted as $P_i^t(j)$. Note we use $L_i^t(j) = 0$ to denote task j is not assigned to worker i at step t, and naturally under this case $P_i^t(j) = 0$. After every step, the data requester collects labels and aggregates them through Bayesian inference (Zheng et al. 2017). Denote the aggregated labels as $\hat{L}^t(j)$, and we can compute the accuracy as $A_t = \frac{1}{M}\sum_{j=1}^{M} 1(\hat{L}^t(j) = L^t(j))$. The revenue for the data requester can then be computed as

$$r_t = F(A_t) - \eta \sum_{i=1}^{N} \sum_{j=1}^{M} P_i^t(j) \tag{1}$$

where $F(\cdot)$ is a non-decreasing monotonic function mapping accuracy to revenue and η is a tunable parameter balancing label quality and costs. Intuitively, the $F(\cdot)$ function needs to be non-deceasing as higher accuracy is preferred and also labels are only useful when their accuracy reaches a certain requirement. Note our framework is robust to any formulation of $F(\cdot)$ function. If not otherwise specified, $F(\cdot)$ is set as $F(A_t) = A_t^{10}$ in our experiments. Our goal is to maximize the cumulative discounted revenue $R = \sum_{t=1}^{T} \gamma^{(t-1)} r_t$, where $\gamma \approx 1.0$ is the discount rate and T is the ending time, by making wise choices of actions (i.e. deciding the offered scaling levels) at each step.

Reinforcement Peer Prediction

We present *Reinforcement Peer Prediction* in Figure 1. Note at every step t, the payment to worker i for task j is factored as a function over two elements, namely $P_i^t(j) = I_t \cdot p_i^t(j)$, where $I_t \geq 0$ is the scaling level decided by our RL algorithm, and $p_i^t(j)$ is the output of our peer prediction scoring function. Here, we assume that the available scaling levels are limited and they form a set W. Thus, the action space of our RL algorithm is W and it needs to learn the optimal policy to adjust the scaling level $I_t \in W$ so that the data requester's revenue R can be maximized.

Since the ground-truth is not available, we cannot directly compute the reward (i.e accuracy A_t), following each action (i.e. deciding the offered scaling level). Following MAP (maximum a posteriori estimation) principle, we resort to the expected value $\mathbb{E}A_t$ as an unbiased estimator of the real accuracy A_t. It can be calculated as

$$\mathbb{E}A_t = \frac{1}{M} \sum_{j=1}^{M} \Pr(\hat{L}^t(j) = L^t(j)) \tag{2}$$

where $\hat{L}^t(j)$s and $L^t(j)$s are the aggregate and true labels at step t, respectively. Besides used to aggregate data, Bayesian inference is also used to generate confusion matrices of all workers and the distribution of task labels $[\Pr(L = 1), \Pr(L = 2)]$, with again L denoting the ground-truth label.

For worker i, his confusion matrix $C_i = [c_{ikg}]_{2\times 2}$ is a measurement of how much effort he exerts, where c_{ikg} denotes the probability that worker i labels a task in class k as class g. Each worker can chose to exert either High effort or Low effort. Denote this decision variable as e_i. Exerting high effort leads a more accurate confusion matrix (higher values on diagonal), but it will also incur a cost $b > 0$.

Since the accuracy of labels is determined by the overall effort of workers, we denote the state of the whole worker crowd s_t by workers' average probability of being correct, namely

$$s_t = \frac{1}{N} \cdot \sum_{k=1}^{2} \Pr(L = k) \cdot \sum_{i=1}^{N} c_{ikk}. \tag{3}$$

After receiving the payment, workers may potentially adjust their strategies in exerting effort based on their rationality model, leading to the change of their state s_t. However, when deciding the next-step scaling level I_{t+1}, we can only estimate the last step state s_t. In other words, the state observation in our mechanism is not only imperfect, but also one-step delayed. This also makes our reinforcement learning problem different from traditional ones.

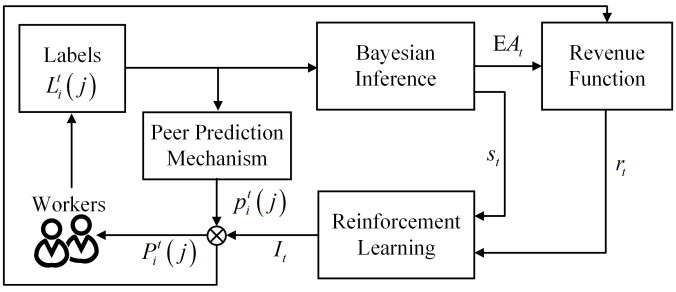


Figure 1: Illustration of our mechanism

Peer Prediction Mechanism

We adopt the multi-task mechanism proposed by Dasgupta and Ghosh (2013). For each worker-task pair (i, j), it selects a reference worker k. Suppose workers i and k have been assigned d other distinct tasks $\{i_1, \ldots, i_d\}$ and $\{k_1, \ldots, k_d\}$,

respectively. Then, the payment to worker i after he labels task j as $L_i^t(j)$ computes as

$$p_i^t(j) = \mathbb{1}[L_i^t(j) = L_k^t(j)] - \xi_i^d \cdot \xi_k^d - \bar{\xi}_i^d \cdot \bar{\xi}_k^d \quad (4)$$

where $\xi_i^d = \sum_{g=1}^{d} \mathbb{1}(L_k^t(i_g) = 1)/d$ and $\bar{\xi}_i^d = 1 - \xi_i^d$.

We adopt above specific mechanism mainly for two reasons. First is due to its simple and prior-free implementation. Second is due to the fact that uninformative equilibrium where agents all report the same labels is a bad equilibrium under this mechanism.

Bayesian Inference

Suppose the prior distributions follow:

$$c_{ik1} \sim \text{Beta}(\alpha_{ik1}^p, \alpha_{ik2}^p)$$

and

$$\Pr(L = 1) \sim \text{Beta}(\beta_1^p, \beta_2^p)$$

Then, we can explicitly derive the posterior distribution of the true labels of the assigned tasks, using the collected labels from workers, as follows:

$$\Pr\left(L^t(1), \ldots, L^t(M) | \mathcal{L}^t\right) \propto B(\boldsymbol{\beta}^t) \prod_{i=1}^{N} \prod_{k=1}^{K} B(\boldsymbol{\alpha}_{ik}^t)$$
$$\alpha_{ikg}^t = \sum_{j=1}^{M} \delta_{ijg}^t \xi_{jk}^t + \alpha_{ikg}^p, \beta_k^t = \sum_{j=1}^{M} \xi_{jk}^t + \beta_k^p \quad (5)$$

where $\mathcal{L}^t$ denotes the set of all collected labels and $B(\cdot)$ denotes the beta function, $\delta_{ijg}^t = \mathbb{1}(L_i^t(j) = g)$ and $\xi_{jk}^t = \mathbb{1}(L^t(j) = k)$. According to Gibbs sampling, we can generate posterior samples via iteratively sampling the following conditional probability distribution

$$\Pr\left(L^t(j) | \mathcal{L}^t, L^t(s \neq j)\right) \propto \Pr\left(L^t(1), \ldots, L^t(M) | \mathcal{L}^t\right). \quad (6)$$

After generating W posterior samples

$$\{(L^{t(1)}(1), \ldots, L^{t(1)}(M)), \ldots, (L^{t(1)}(1), \ldots, L^{t(1)}(M))\},$$

we can calculate the posterior distribution estimation for the true label of task j as

$$\Pr\left(L^t(j) = k\right) = \frac{1}{W} \sum_{s=1}^{W} 1\left(L^{t(s)}(j) = k\right). \quad (7)$$

Meanwhile, we can decide the aggregated label of task j as

$$\hat{L}^t(j) = \arg\max_{k \in \{1,2\}} \Pr(L^t(j) = k). \quad (8)$$

We can also estimate worker i's confusion matrix as

$$c_{ikg}^t = \frac{\alpha_{ikg}^p + \sum_{s=1}^{W} \sum_{j=1}^{M} \mathbb{1}(L^{t(s)}(j) = k) \cdot \delta_{ijg}^t}{\sum_{q=1}^{2} \alpha_{ikq}^p + \sum_{s=1}^{W} \sum_{j=1}^{M} \mathbb{1}(L^{t(s)}(j) = k)}. \quad (9)$$

The above jointly help us track the state of workers.

Reinforcement Learning

Recall that when computing the scaling level I_t for step t, we do not observe workers' labels and thus cannot estimate the current state s_t via Bayesian inference. In this case, we cannot build the scaling level adjustment policy by directly using the the learned state of the whole workers. Meanwhile, the number of labels the data requester collected at one time is usually very limited. If there are still many unlabeled tasks, we may need to use very high scaling levels to impress workers with the importance of high effort. On the other hand, if there are only few tasks left, we may tend to using low scaling levels to reduce the payment. Thus, our scaling level adjustment policy should take the number of unlabeled tasks into consideration. To define our policy, we firstly introduce the augmented state of our mechanism as

$$\hat{s}_t = \langle s_{t-1}, I_{t-1}, T - t \rangle. \quad (10)$$

where s_{t-1} and I_{t-1} reflect the current state, and $T - t$ can denote the number of unlabeled tasks because we collect labels for a fixed number of tasks at each step. Then, our task adjustment policy $\pi(I_t | \hat{s}_t)$. Typically, in model-free reinforcement learning methods, Q-function, which calculates the expected reward of taking a given action in a given state and following the current policy afterwards, is heavily used to iteratively improve the adopted policy. Puting into the context of our problem, the Q-function of our scaling level adjustment policy can be calculated as

$$Q(\hat{s}_t, I_t) = \sum_{i=0}^{T-t} \gamma^i r_{t+i}. \quad (11)$$

Then, our policy can be formally written as

$$I_t = \arg\max_{I \in W} Q(\hat{s}_t, I_t). \quad (12)$$

Another challenge of our mechanism is that both the state s_t and reward r_t can not be accurately observed. Thus, we approximate the temporal difference (TD) by assuming that the residual follows a Gaussian process:

$$r_t \approx Q(\hat{s}_t, I_t) - \gamma Q(\hat{s}_{t+1}, I_{t+1}) + N(\hat{s}_t, \hat{s}_{t+1}) \quad (13)$$

where $N(\hat{s}_t, \hat{s}_{t+1}) \sim \mathcal{N}(0, \sigma^2)$ approximates the residual. Here, we use the Gaussian distribution to approximate the residual because of two reasons. Firstly, the Gaussian distribution brings much convenience for our derivation. Secondly, our empirical results later show that this approximation has achieved very good performance on different worker models. Under the Gaussian process approximation, we can put all the observed rewards and the corresponding Q-function up to the current step t together and obtain

$$\boldsymbol{r} = \boldsymbol{H}\boldsymbol{Q} + \boldsymbol{N} \quad (14)$$

where $\boldsymbol{r}$, $\boldsymbol{Q}$ and $\boldsymbol{N}$ denote the collection of rewards, Q values, and residual values up to step t, respectively. Due to the Gaussian process assumption of the residual, $\boldsymbol{N} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\sigma}^2)$, where $\boldsymbol{\sigma}^2 = \text{diag}(\sigma^2, \ldots, \sigma^2)$. The hat matrix $\boldsymbol{H}$ satisfies that $\boldsymbol{H}(k, k) = 1$ and $\boldsymbol{H}(k, k+1) = -\gamma$ for $k = 1, \ldots, t$. Then, the Q-function can be learned effectively using the online Gaussian process regression algorithm (Engel, Mannor, and Meir 2005). Furthermore, we use

the classic ϵ-greedy policy to learn the Q-function. In more specifics, with a probability of $1-\epsilon$, $\arg\max_{I_t} Q(\hat{s}_t, I_t)$ is chosen; with a probability of ϵ, a random I_t is used. Note the optimal policy learned by our mechanism factors in the ϵ influence, and we also do not turn it off during evaluation.

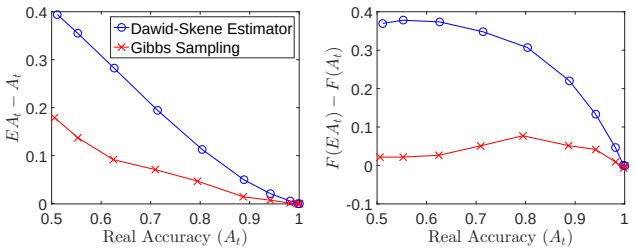


Figure 2: Mean errors of accuracy estimations in 200 runs

Experiments

Figure 3 shows our experimental results on three popular worker models. Similar to the previous studies (Liu and Chen 2017), we assume that there are two effort levels, High and Low, that a worker can potentially choose from. If a worker choose the high effort, it will report the correct label with probability 0.9; otherwise, it will randomly report the label — i.e. the probability of being correct is 0.5. Then, the three worker models can be described as

- **Rational Workers**: Since the peer prediction mechanism can theoretically ensure that exerting the maximal effort is the optimal action for all workers, we assume that rational workers will choose the high effort level no matter which scaling level is provided by our mechanism.
- **Quantal Response Workers** (McKelvey and Palfrey 1995): The quantal response worker will choose the high level of effort with the following probability

$$\Pr(H) = \frac{\exp(\lambda \cdot u_H)}{\exp(\lambda \cdot u_H) + \exp(\lambda \cdot u_L)} \tag{15}$$

  where u_H and u_L denote workers' expected utility of exerting high and low effort, respectively. λ denotes workers' rationality level, and we set $\lambda = 3$ in our experiments.
- **MWU Workers** (Chastain et al. 2014): The multiplicative weight update (MWU) workers uses a mixed strategy $[\Pr_t(H), \Pr_t(L)]$ to decide their effort levels at step t. After receiving the payment from our mechanism, they will update their strategy as

$$\begin{aligned} \Pr_{t+1}(H) &= Z \cdot \Pr_t(H) \cdot (1 + \bar{u}_H) \\ \Pr_{t+1}(L) &= Z \cdot \Pr_t(L) \cdot (1 + \bar{u}_L) \end{aligned} \tag{16}$$

  where Z is the normalizing constant used to keep the probability sum being 1. $\bar{u}_H$ and $\bar{u}_L$ are the average utility of high and low effort, respectively.

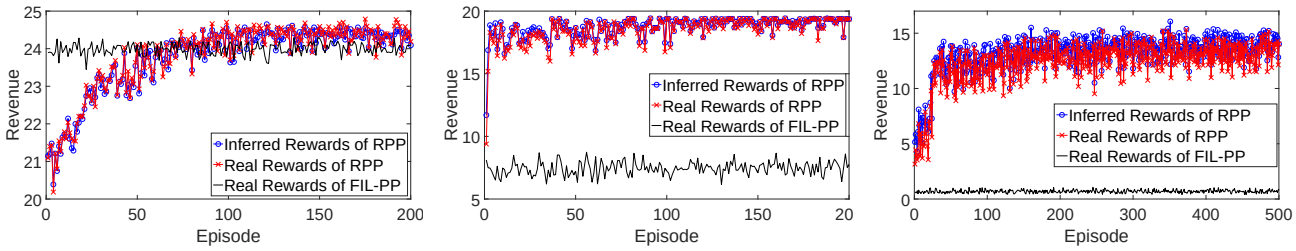
Note that we only model agents using existing bounded rationality models here. More sophisticated worker model, e.g. the forward-looking ones, will be studied in future work.

Furthermore, in our experiments, we assume there are four available scaling levels for our mechanism to choose, namely $I_t \in W = \{0.1, 1.0, 5.0, 10.0\}$. In practice, traditional one-shot peer prediction mechanisms are often implemented with a fixed scaling level. Because of this, we set up a mechanism with a fixed scaling level of 1.0 as baseline and we denote it as FIL-PP. By contrast, our reinforcement peer prediction (RPP) mechanism can dynamically learn and adjust the scaling level to maximize data requester's revenue. Based on all experiment results, we find that our mechanism is robust, and is able to significantly increase data requester's revenue, especially when workers are not fully rational. Besides, in Figure 2, we present the comparison of accuracy errors between the classic Dawid-Skene estimator (Dawid and Skene 1979) and our Gibbs sampling estimator. The results show that our estimator can significantly reduce the errors of A_t and $F(A_t)$ estimations. The accurate estimation of $F(A_t)$ then can provide a high-quality reward signal, which warrants the good performance of our closed-loop reinforcement framework.

Conclusions and Future Work

In this paper, we propose a novel model-free reinforcement peer prediction mechanism. It maximizes the revenue of the data requester by dynamically adjusting the scaling level based on workers' labels. Compared with existing peer prediction mechanisms, our mechanism does not require to assume a decision-making model for workers and removes the uninformative equilibrium where agents report uninformative information. To achieve these advantages, we firstly develop a Bayesian inference algorithm to evaluate workers' contributions, and to infer the reward following each offered scaling level. Then, we learn the optimal scaling level adjustment strategy by approximating the Q-function of our mechanism with the Gaussian process. We conduct empirical evaluation on three popular worker models, and the results show that our mechanism is robust, and is able to significantly increase data requester's revenue.

In future, we wish to improve our mechanism in the following two aspects. Firstly, more complex worker models and actions should be considered. To the best of our knowledge, this paper is the first work to adjust the peer prediction mechanism based workers' labels. In this case, except for the existing bounded rationality models, we cannot find models to depict workers' more complex behaviors, e.g. forward-looking. Meanwhile, our interaction with workers is limited to the scaling level, while the practical crowdsouricng platform can affect workers by, for example, sending them an alert or forcing them to pass a training process. Currently, we can not find proper models to depict workers' response to these actions. Secondly, we can improve the convergence speed of our reinforcement learning algorithm by introducing more complex learning structures, for example, the actor-critic algorithm. Besides, we also consider to build a deep reinforcement peer prediction mechanism which directly uses the collected labels as the state. By doing so, we can avoid using the Bayesian inference algorithm which assumes a fixed confusion matrix for one worker.



(a) Fully Rational Workers (b) Quantal Response Workers (c) MWU Workers

Figure 3: Experiments on three popular worker models

Acknowledgments

This work was conducted within Rolls-Royce@NTU Corporate Lab with support from the National Research Foundation (NRF) Singapore under the Corp Lab@University Scheme. The authors also thank Anxiang Zeng from Alibaba Group for valuable discussions.

References

Chastain, E.; Livnat, A.; Papadimitriou, C.; and Vazirani, U. 2014. Algorithms, games, and evolution. *PNAS* 111(29):10620–10623.

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

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

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *Proc. of CVPR*.

Engel, Y.; Mannor, S.; and Meir, R. 2005. Reinforcement learning with gaussian processes. In *Proc. of ICML*.

Gao, X. A.; Mao, A.; Chen, Y.; and Adams, R. P. 2014. Trick or treat: putting peer prediction to the test. In *Proceedings of the fifteenth ACM conference on Economics and computation*, 507–524. ACM.

Jurca, R., and Faltings, B. 2007. Robust incentive-compatible feedback payments. In *Agent-Mediated Electronic Commerce. Automated Negotiation and Strategy Design for Electronic Markets*. Springer. 204–218.

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

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

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

Miller, N.; Resnick, P.; and Zeckhauser, R. 2005. Eliciting informative feedback: The peer-prediction method. *Management Science* 51(9):1359–1373.

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

Shnayder, V.; Agarwal, A.; Frongillo, R.; and Parkes, D. C. 2016. Informed truthfulness in multi-task peer prediction. In *Proc. of ACM EC*.

Snow, R.; O'Connor, B.; Jurafsky, D.; and Ng, A. Y. 2008. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proc. of EMNLP*.

Waggoner, B., and Chen, Y. 2014. Output agreement mechanisms and common knowledge. In *Proc. of HCOMP*.

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

Witkowski, J., and Parkes, D. C. 2012b. A robust bayesian truth serum for small populations. In *Proc. of AAAI*.

Witkowski, J.; Bachrach, Y.; Key, P.; and Parkes, D. C. 2013. Dwelling on the negative: Incentivizing effort in peer prediction. In *Proc. of HCOMP*.

Zheng, Y.; Li, G.; Li, Y.; Shan, C.; and Cheng, R. 2017. Truth inference in crowdsourcing: is the problem solved? *Proc. of the VLDB Endowment* 10(5):541–552.