Redesign Peer Prediction with Machine Learning

 # 123

Abstract

...

1 Motivation

Peer prediction mechanisms have two fatal drawbacks:

- Existing peer prediction mechanisms only care about incentive compatibility (IC) which only poses requirements to the expected incentives to workers. They achieve IC via comparing the reports between the targeted and selected reference agents. In this way, they only use a tiny part of the information behind all collected labels. Besides, they never analyze the stochastic property of incentives and the variation of incentives among different types of agents.

- 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 is strong evidence showing that human workers are not always fully rational, and they may deviate from equilibrium strategies. Thus, these peer prediction mechanisms which is fancy in theory may yet fail in practice.

2 Contribution

We have two core contributions in this paper:

- We propose a novel one-shot peer prediction mechanism based on Bayesian inference. Since existing Bayesian inference algorithms (e.g. EM estimator and variational inference) for crowdsourcing are biased in principle, we derive the explicit posterior distribution of the true labels and employ Gibbs sampling for inference. The most challenging problem of our mechanism is to prove the incentive compatibility of our mechanism which has never been explored in the literature. Besides, we also empirically show the advantages of our mechanism on the stability and fairness of incentives over existing ones.

- We design the first reinforcement peer prediction framework which sequentially interacts with workers. It dynamically adjusts the scaling level of our peer prediction mechanism to maximize the utility of the data requester.

To avoid assuming a decision-making model for workers, we use the data-driven Gaussian process to represent the scaling level adjustment policy, and online updates our policy according to workers' responses. We theoretically prove the incentive compatibility of our framework and empirically show its advantages on improving the utility if the data requester over one-shot mechanisms.

3 One-Shot Peer Prediction Mechanism

3.1 Formulation and settings

Suppose in our system there is one data requester who assigns M tasks with answer space $\{1, 2\}$ to $N \geq 3$ candidate workers. We denote all tasks and workers by $\mathcal{T} = \{1, 2, \ldots, M\}$ and $\mathcal{C} = \{1, 2, \ldots, N\}$, respectively. The label $L_i(j)$ generated by worker $i \in \mathcal{C}$ for task $j \in \mathcal{T}$ comes from a distribution that depends both on the ground-truth label $L(j)$ and worker i's effort level e_i and reporting strategy r_i. Suppose there are two effort levels, High ($e_i = 1$) and Low ($e_i = 0$), that a worker can potentially choose from. Meanwhile, worker i can decide either to truthfully report his observation $r_i = 1$ or to revert the answer $r_i = 0$. Note that worker i's effort level and reporting strategy may be a mixed of the above pure actions. Thus, we use $e_i \in [0, 1]$ and $r_i \in [0, 1]$ to denote worker i's probability of exerting high efforts and being truthful, respectively. Using the above notations, we can calculate worker i's probability of providing the correct label as

$$p_i = r_i e_i p_{i,H} + r_i(1 - e_i)p_{i,L} + \\ (1 - r_i)e_i(1 - p_{i,H}) + (1 - r_i)(1 - e_i)p_{i,L} \quad (1)$$

where $p_{i,H}$ and $p_{i,L}$ denote the probability of observing the correct label when worker i exerts high and low efforts, respectively. Following the previous studies on peer prediction [Dasgupta and Ghosh, 2013], we assume that $p_{i,H} > p_{i,L} \geq 0.5$. Then, we can easily obtain Theorem 1.

Theorem 1. p_i *is strictly increasing with* e_i *and* r_i.

3.2 Bayesian inference

The joint distribution of all the collected labels $\mathcal{L} = [L_i(j)]$ and the true labels $\boldsymbol{L} = [L(1), \ldots, L(M)]$ satisfies

$$P(\mathcal{L}, \boldsymbol{L} | \boldsymbol{p}, \boldsymbol{\tau}) = \\ \prod_{j=1}^{M} \prod_{k=1}^{K} \left\{ \tau_k \prod_{i=1}^{N} p_i^{\delta_{ijk}}(1 - p_i)^{\delta_{ij(3-k)}} \right\}^{\xi_{jk}} \quad (2)$$

where $\boldsymbol{p} = [p_i]_N$ and $\boldsymbol{\tau} = [\tau_1, \tau_2]$. τ_1 and τ_2 denote the distribution of answer 1 and 2 among all tasks, respectively. Besides, $\delta_{ijk} = \mathbb{1}(L_i(j) = k)$ and $\xi_{jk} = \mathbb{1}(L^t(j) = k)$. Referring to the literature on Bayesian inference [Liu *et al.*, 2012], we assume a Dirichlet prior for both p_i and $\boldsymbol{\tau}$ as

$$[p_i, 1 - p_i] \sim \text{Dir}(\alpha_1, \alpha_2) \;, \; \boldsymbol{\tau} \sim \text{Dir}(\beta_1, \beta_2). \quad (3)$$

where $\text{Dir}(\cdot)$ denotes the Dirichlet distribution. Then, we can derive the joint distribution of $\mathcal{L}$, $\boldsymbol{L}$, $\boldsymbol{p}$ and $\boldsymbol{\tau}$ as

$$P(\mathcal{L}, \boldsymbol{L}, \boldsymbol{p}, \boldsymbol{\tau} | \boldsymbol{\alpha}, \boldsymbol{\beta}) = P(\mathcal{L}, \boldsymbol{L} | \boldsymbol{p}, \boldsymbol{\tau}) \cdot P(\boldsymbol{p}, \boldsymbol{\tau} | \boldsymbol{\alpha}, \boldsymbol{\beta})$$

$$= \frac{1}{B(\boldsymbol{\beta})} \prod_{k=1}^{K} \tau_k^{\hat{\beta}_k - 1} \cdot \prod_{i=1}^{N} \frac{1}{B(\boldsymbol{\alpha})} p_i^{\hat{\alpha}_{i1} - 1} (1 - p_i)^{\hat{\alpha}_{i2} - 1} \quad (4)$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2]$, $\boldsymbol{\beta} = [\beta_1, \beta_2]$ and

$$\hat{\alpha}_{i1}^t = \sum_{j=1}^{M} \sum_{k=1}^{K} \delta_{ijk}^t \xi_{jk}^t + \alpha_1$$

$$\hat{\alpha}_{i2}^t = \sum_{j=1}^{M} \sum_{k=1}^{K} \delta_{ij(3-k)}^t \xi_{jk}^t + \alpha_2 \quad (5)$$

$$\hat{\beta}_k^t = \sum_{j=1}^{M} \xi_{jk}^t + \beta_k.$$

$B(\cdot)$ denotes the beta function which satisfies

$$B(x, y) = \frac{(x-1)!(y-1)!}{(x+y-1)!}. \quad (6)$$

Furthermore, we can conduct marginalization via integrating Equation 4 over all possible values of $\boldsymbol{p}$ and $\boldsymbol{\tau}$ as

$$P(\mathcal{L}, \boldsymbol{L} | \boldsymbol{\alpha}, \boldsymbol{\beta}) = \int_{\boldsymbol{p}, \boldsymbol{\tau}} P(\mathcal{L}, \boldsymbol{L}, \boldsymbol{p}, \boldsymbol{\tau} | \boldsymbol{\alpha}, \boldsymbol{\beta}) \mathrm{d}\boldsymbol{p} \mathrm{d}\boldsymbol{\tau}$$

$$= \frac{B(\hat{\boldsymbol{\beta}})}{B(\boldsymbol{\beta})} \cdot \prod_{i=1}^{N} \frac{B(\hat{\boldsymbol{\alpha}}_i)}{B(\boldsymbol{\alpha})} \quad (7)$$

where $\hat{\boldsymbol{\alpha}}_i = [\alpha_{i1}, \alpha_{i2}]$ and $\hat{\boldsymbol{\beta}} = [\hat{\beta}_1, \hat{\beta}_2]$. Following Bayes' theorem, we can know the posterior distribution satisfies

$$P(\boldsymbol{L} | \mathcal{L}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{P(\mathcal{L}, \boldsymbol{L} | \boldsymbol{\alpha}, \boldsymbol{\beta})}{P(\mathcal{L} | \boldsymbol{\alpha}, \boldsymbol{\beta})} \propto B(\hat{\boldsymbol{\beta}}) \prod_{i=1}^{N} B(\hat{\boldsymbol{\alpha}}_i). \quad (8)$$

Note that the previous studies have shown that we should be optimistic about workers' willingness to provide the correct label [Chen *et al.*, 2015], which requires $\alpha_1 > \alpha_2$. In this paper, for the simplicity of derivation, we keep $\alpha_1 = 2$ and $\alpha_2 = 1$. Besides, since we have no knowledge about the distribution of answer 1 and 2, we employ the non-informative uniform distribution for $\boldsymbol{\tau}$, that is, setting $\beta_1 = \beta_2 = 1$.

To observe the posterior distribution, we resort to the classic Gibbs sampling. Firstly, according to Bayes' theorem, we can know the conditional posterior distribution satisfies

$$P(L(j) | \mathcal{L}, \boldsymbol{L}(\bar{j}), \boldsymbol{\alpha}, \boldsymbol{\beta}) \propto P(\boldsymbol{L} | \mathcal{L}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \quad (9)$$

where $\boldsymbol{L}(\bar{j})$ denotes the true labels of all tasks expect for task j. Then, we can generate the samples of the posterior distribution $P(\boldsymbol{L} | \mathcal{L}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ by using Algorithm 1. In each round of sampling, Algorithm 1 traverses all tasks by increasing j from

---

Algorithm 1: Gibbs sampling for crowdsourcing

Input: the collected labels $\mathcal{L}$, the number of samples W
Output: the sample sequence $\mathcal{S}$

1   $\mathcal{S} \leftarrow \emptyset$, Initialize $\boldsymbol{L} = [L(j)]_M$ with the uniform distribution
2   **for** $s = 1$ **to** W **do**
3   **for** $j = 1$ **to** M **do**
4   Set $L(j) = 1$ and compute $x_1 = B(\hat{\boldsymbol{\beta}}) \prod_{i=1}^{N} B(\hat{\boldsymbol{\alpha}}_i)$
5   Set $L(j) = 2$ and compute $x_2 = B(\hat{\boldsymbol{\beta}}) \prod_{i=1}^{N} B(\hat{\boldsymbol{\alpha}}_i)$
6         $L(j) \leftarrow$ Sample $\{1, 2\}$ with $P(1) = x_1/(x_1 + x_2)$
7   Append $\boldsymbol{L}$ to the sample sequence $\mathcal{S}$

---

1 to M and always update the true label vector $\boldsymbol{L}$ via replacing $L(j)$ with the newly obtained sample (line 3-6). Here, we write the s-th sample as $\boldsymbol{L}^{(s)}$. Since Gibbs sampling requires the burn-in process, we need to discard the first b samples in the obtained sample sequence $\mathcal{S}$ and can only use the latter $W - b$ samples. Thus, the marginal distribution of the true label of task j can be approximately calculate as

$$P(L(j) = k) = \frac{1}{W - b} \sum_{s=b+1}^{W} \mathbb{1}\left(L^{s)}(j) = k\right). \quad (10)$$

Following the maximum a posteriori probability rule, we can decide the aggregated label of task j as

$$\hat{L}^t(j) = \arg\max_{k \in \{1, 2\}} \Pr(L^t(j) = k). \quad (11)$$

Besides, we can estimate worker i's probability to provide the correct label as

$$\hat{p}_i = \frac{\sum_{s=b+1}^{W} \left[\alpha_{i1} + \sum_{j=1}^{M} \mathbb{1}(L^{(s)}(j) = L_i(j)) \right]}{(W - b) \cdot (\alpha_{i1} + \alpha_{i2} + M)}. \quad (12)$$

Note that both W and b should be large values, and in this paper, we set $W = 1000$ and $b = 100$.

3.3 Bayesian peer prediction mechanism

Now, we present the definition of our mechanism as follows:

Definition 1. *The Bayesian peer prediction mechanism computes worker i's rewards for her reports on M tasks as:*

$$R(i) = M \cdot [a \cdot (\hat{p}_i - 0.5) + b] \quad (13)$$

where $a > 0$ and $b \geq 0$ are the scaling level and the guaranteed base payment, respectively.

To prove the incentive compatibility of our mechanism, we firstly study the convergence of our Bayesian inference algorithm and can obtain Theorem 3.

Theorem 2. *Let $n(p_i > 0.5)$ and $n(p_i < 0.5)$ be the number of workers who are better and worse than random guess ($p_i = 0.5$), respectively. Then,*

$$\lim_{M \to \infty} |\mathbb{E}_{\mathcal{L}} \hat{p}_i - p_i| \begin{cases} = 0 & n(p_i > 0.5) > n(p_i < 0.5) \\ > 0 & n(p_i > 0.5) \leq n(p_i < 0.5) \end{cases}$$

where the subscript of $\mathbb{E}$ denotes the random variable.

Proof. By symmetry, we can assume the real true labels of all M tasks as 1. For the posterior variable $\boldsymbol{L}$ in the posterior

distribution $P(\boldsymbol{L}|\mathcal{L}, \boldsymbol{\alpha}, \boldsymbol{\beta})$, we introduce n and m to denote the number of 1s and 2s in $\boldsymbol{L}$. Among the n tasks of which the posterior true label is 1, we define x_i and y_i to denote the number of tasks of which the label provided by worker i is 1 and 2, respectively. Among the m tasks of which the posterior true label is 2, we define w_i and z_i to denote the number of tasks of which the label provided by worker i is 1 and 2, respectively. Based on the above definitions, we can know that $n + m = M$, $x_i + y_i = n$ and $w_i + z_i = m$. Besides, for the simplicity of notation, we use $P(\boldsymbol{L})$ to denote the posterior distribution in our following derivation.

The real true labels are 1 while worker i only label $x_i + w_i$ tasks as 1. Thus, the accuracy of worker i is $(x_i + w_i)/M$, and we can have

$$\frac{\mathbb{E}_{\mathcal{L}\sim\boldsymbol{p}}(x_i + w_i)}{M} = p_i. \tag{14}$$

Meanwhile, $\mathbb{1}(L^{(s)}(j) = L_i(j)) = x_i + z_i$. Thus, the inferred probability of being correct in Equation 12 satisfies

$$\lim_{M\to\infty}\hat{p} \approx \lim_{M\to\infty}\frac{\mathbb{E}_{\boldsymbol{L}\sim P(\boldsymbol{L})}(x_i + z_i)}{M}. \tag{15}$$

Thus, the error of our inference can be calculated as

$$\begin{aligned}
\lim_{M\to\infty}|\mathbb{E}_{\mathcal{L}}\hat{p}_i - p_i| &\approx \lim_{M\to\infty}\frac{\mathbb{E}_{\mathcal{L}\sim\boldsymbol{p},\boldsymbol{L}\sim P(\boldsymbol{L})}|w_i - z_i|}{M} \\
&\leq \lim_{M\to\infty}\frac{\mathbb{E}_{\mathcal{L}\sim\boldsymbol{p},\boldsymbol{L}\sim P(\boldsymbol{L})}[m]}{M}.
\end{aligned} \tag{16}$$

To compute the expectation, we need to analyze the probability density function of m. According to Equation 8, we can know the probability density function $P(m)$ satisfies

$$\begin{aligned}
P(m) &= \frac{C_M^m B(n+1, m+1)}{Z}\prod_{i=1}^{N}B(n+2, m+1) \\
&= \frac{1}{Z(M+1)}\prod_{i=1}^{N}B(n+2, m+1) \\
&= \frac{2^{-N(M+1)}}{Z(M+1)}\cdot\exp\left[\sum_{i=1}^{N}F(m; p_i)\right]
\end{aligned} \tag{17}$$

where Z is the normalization constant and

$$\begin{aligned}
F(m; p_i) = \log\Big\{&\sum_{x_i=0}^{n}\sum_{w_i=0}^{m}2^{M+1}C_n^{x_i}C_m^{w_i}\times \\
&p_i^{x_i+w_i}(1-p_i)^{y_i+z_i}B(x_i+z_i+2, y_i+w_i+1)\Big\}
\end{aligned} \tag{18}$$

An interesting property of $F(m; p_i)$ is the symmetry that $F(m; p_i) = F(M - m; 1 - p_i)$. This property can be proved by exchanging x_i and z_i as well as y_i and w_i in Equation 18. Thus, we only need to study the cases where $p_i \geq 0.5$. In Figure 1a, we draw the curves of $F(m)$ corresponding to different values of p and M. From these curves, we can find that, $F(m)$ is monotonically decreasing when m/M increases from 0 to around 0.5, while it is monotonically increasing when m/M increases from around 0.5 to 1. Besides, the value differences of $F(m)$ between the middle region and

the two ends will increase significantly as M and p increase. Meanwhile, we can compute the probability ratio as

$$\frac{P(m)}{P(0)} = \exp\left[\sum_{i=1}^{N}\left(F(m; p_i) - F(0; p_i)\right)\right] \tag{19}$$

which will be very small for the middle region in Figure 1a. In this case, the probability of the middle region will be very small, resulting in a neglectable effects on the expectation. Thus, as shown in Figure 1b, we set up two linear functions to accurately approximate the complex $F(m; p_i)$ at two ends. As for the middle region, we actually do not care whether the approximation is accurate or not. Then, we can have

$$F(m; p_i) \approx \begin{cases} F(0; p) - k_0 m & m \leq 0.5M \\ F(M; p) - k_1(M - m) & m > 0.5M \end{cases}$$

where the parameters $F(0; p)$, k_0, $F(M; p)$ and k_1 need to be determined by further analysis on $F(m; p_i)$.
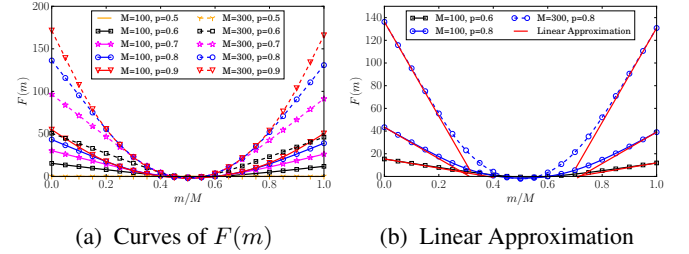

(a) Curves of $F(m)$ (b) Linear Approximation

Figure 1: Empirical study on $F(m)$

$\square$

Next, we compute $F(0; p)$ as follows:

$$\begin{aligned}
\frac{e^{F(0;p)}}{2^{M+1}} &= \sum_{x=0}^{M}C_M^x p^x(1-p)^y B(x+2, y+1) \\
&= \frac{(1-p)^M}{(M+1)(M+2)}\sum_{x=0}^{M}(x+1)\lambda^x \\
&= \frac{(1-p)^M}{(M+1)(M+2)}\cdot\frac{\mathrm{d}}{\mathrm{d}\lambda}\sum_{x=0}^{M}\lambda^{x+1} \\
&= \frac{(1-p)^{M+2} - (M+2)(1-p)p^{M+1} + (M+1)p^{M+2}}{(M+1)(M+2)(1-2p)^2}
\end{aligned}$$

where $\lambda = p/(1-p)$. When M is very large, we can have the following approximation:

- When $p > 0.5$,
$$F(0; p) \approx (M+1)\log(2p) - \log(M) - \log|2p - 1|.$$

- When $p = 0.5$, $F(0; p) = 0$.

- When $p < 0.5$,
$$\begin{aligned}
F(0; p) \approx &(M+1)\log(2 - 2p) \\
&- 2\log(M) + \log(1-p) - 2\log|2p - 1|.
\end{aligned}$$

Considering the symmetry of $F(m; p)$, we can have

- When $p > 0.5$,

$$F(M;p) \approx (M+1)\log(2p) - 2\log(M) + \log(1-p) - 2\log|2p-1|.$$

- When $p = 0.5$, $F(0;p) = 0$.
- When $p < 0.5$,

$$F(0;p) \approx (M+1)\log(2-2p) - \log(M) - \log|2p-1|.$$

Then, we can have

$$F(0,p) - F(M,p) \approx \begin{cases} \log(M) & p > 0.5 \\ 0 & p = 0.5 \\ -\log(M) & p < 0.5 \end{cases} . \quad (20)$$

Furthermore, we compute $F(1,p)$ as follows:

$$\frac{e^{F(1;p)}}{2^{M+1}} = \sum_{x=0}^{M-1} C_{M-1}^x p^x (1-p)^{y+1} B(x+3, y+1) +$$

$$\sum_{x=0}^{M-1} C_{M-1}^x p^{x+1}(1-p)^y B(x+2, y+2)$$

$$= \frac{p(1-p)^{M-1}}{M(M+1)} \sum_{x=0}^{M-1}(x+1)\lambda^x +$$

$$\frac{(1-2p)(1-p)^{M-1}}{M(M+1)(M+2)} \sum_{x=0}^{M-1}(x+1)(x+2)\lambda^x$$

$$\sum_{x=0}^{M-1}(x+1)\lambda^x = \frac{d}{d\lambda} \sum_{x=0}^{M-1} \lambda^{x+1} = \frac{d}{d\lambda}\left(\frac{\lambda - \lambda^{M+1}}{1-\lambda}\right)$$

$$\sum_{x=0}^{M-1}(x+1)(x+2)\lambda^x = \frac{d^2}{d^2\lambda}\left(\frac{\lambda^2 - \lambda^{M+2}}{1-\lambda}\right)$$

When M is very large, we approximate $F(1;p)$ as:

- When $p > 0.5$,

$$F(1;p) \approx M\log(2p) + \log(2-2p+\frac{2}{M}) - \log(M) - \log|2p-1|.$$

- When $p = 0.5$, $F(1;p) = 0$.
- When $p < 0.5$,

$$F(0;p) \approx (M+1)\log(2-2p) + \log(p+\frac{1}{M}) - 2\log(M) + \log(1-p) - 2\log|2p-1|.$$

Thus, the slope of our left-end linear function in Figure 1b can be computed as

$$k_0 \approx \begin{cases} \log(\lambda) & p > 0.5 \\ 0 & p = 0.5 \\ -\log(\lambda) & p < 0.5 \end{cases} . \quad (21)$$

Using the symmetry of $F(m;p)$, we can prove that the right-end linear function satisfies $k_1 = k_0$.

If we define $\rho = m/M$, then we can have

- When $\rho \leq 0.5$,

$$\sum_{i=1}^{N} F(m;p_i) \approx C - KM\rho \quad (22)$$

- When $\rho > 0.5$,

$$\sum_{i=1}^{N} F(m;p_i) \approx C + \delta - KM(1-\rho) \quad (23)$$

where

$$\delta = [n(p_i < 0.5) - n(p_i > 0.5)]\log(M).$$

Besides, $C = \sum_{i=1}^{N} F(0,p_i)$ and it satisfies

$$\lim_{M\to\infty} C = +\infty. \quad (24)$$

$K = \sum_{i=1}^{N} k_0(p_i) \geq 0$ and the equality holds only when $p_i = 0.5$ holds for all workers. Then, we can compute the expectation as follows:

$$\lim_{M\to\infty} \mathbb{E}\left[\frac{m}{M}\right] = \lim_{M\to\infty} \sum_{m=0}^{M} \frac{m}{M}P(m) = \lim_{M\to\infty} \frac{A_1 + A_2}{B_1 + B_2}$$

where

$$A_1 = \sum_{m=0}^{\lfloor 0.5M \rfloor} \frac{m}{M} e^{C-Km}$$

$$= \frac{e^C}{KM} \frac{d}{du} \sum_{m=0}^{\lfloor 0.5M \rfloor} u^{Km+1} \Bigg|_{u=e^{-1}} - \frac{1}{KM} B_1$$

$$\approx \frac{1+(K+1)e^{-K}}{(1-e^{-K})^2} \cdot \frac{e^C}{KM} - \frac{1}{KM} B_1$$

$$A_2 = \sum_{m=\lceil 0.5M \rceil}^{M} \frac{m}{M} e^{C+\delta-K(M-m)}$$

$$= \sum_{u=0}^{\lfloor 0.5M \rfloor} \left(1 - \frac{u}{M}\right) e^{C+\delta-Ku} = e^\delta(B_1 - A_1)$$

$$B_1 = \sum_{m=0}^{\lfloor 0.5M \rfloor} e^{C-Km} = \left[1 - e^{-0.5KM}\right] e^C/(1-e^{-K})$$

$$B_2 = \sum_{m=\lceil 0.5M \rceil}^{M} e^{C+\delta-K(M-m)} = e^\delta B_1$$

Thus, when $n(p_i > 0.5) > n(p_i < 0.5)$,

$$\lim_{M\to\infty} \mathbb{E}\left[\frac{m}{M}\right] = \lim_{M\to\infty} \frac{1-e^\delta}{1+e^\delta} G + \frac{e^\delta}{1+e^\delta} \quad (25)$$

where

$$G = \frac{1+(K+1)e^{-K}}{KM(1-e^{-K})(1-e^{-0.5KM})} - \frac{1}{KM}.$$

Thus,

$$\lim_{M\to\infty} |\mathbb{E}_{\mathcal{L}}\hat{p}_i - p_i| \leq \lim_{M\to\infty} \frac{\mathbb{E}[m]}{M} = 0. \quad (26)$$

(a) Changing the true label distribution   (b) Changing the opponent workers   (c) Changing the targeted worker
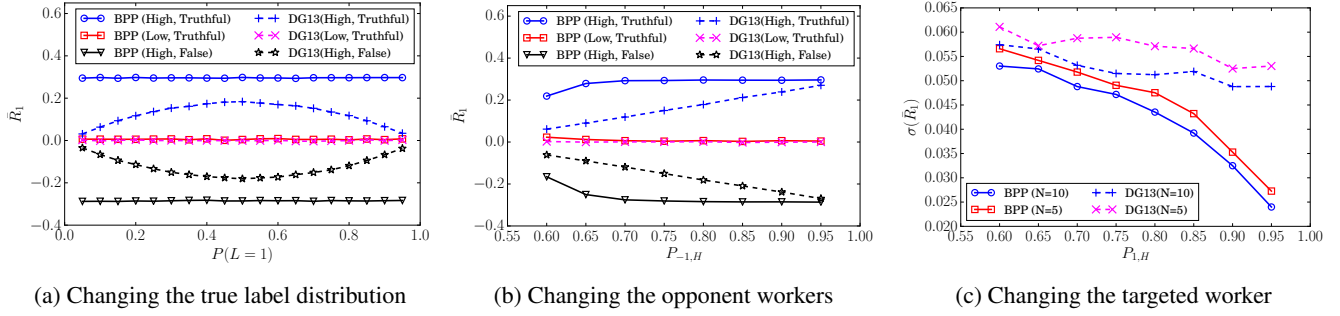
Figure 2: Empirical analysis on the incentive received by worker 1

On the other hand, when $n(p_i > 0.5) \leq n(p_i < 0.5)$,

$$
\begin{aligned}
\lim_{M \to \infty} |\hat{p}_i - p_i| &\approx \lim_{M \to \infty} \frac{\mathbb{E}|w_i - z_i|}{M} \\
&= \lim_{M \to \infty} \mathbb{E}\rho \left| 1 - \frac{2z_i}{m} \right| \\
&\geq \lim_{M \to \infty} \sum_{z_i=0}^{M} \left| 1 - \frac{2z_i}{M} \right| P(z_i|m=M)P(m=M) \\
&= \lim_{M \to \infty} \sum_{z_i=0}^{M} \left| 1 - \frac{2z_i}{M} \right| P(z_i|m=M)\frac{e^{C+\delta}}{B_1 + B_2}
\end{aligned}
\tag{27}
$$

where

$$
\lim_{M \to \infty} \frac{e^{C+\delta}}{B_1 + B_2} = \frac{1 - e^K}{1 + e^\delta} > 1 - e^K.
$$

Besides, $\left| 1 - \frac{2z_i}{M} \right| \geq 0$ and the equality only holds when $z_i = 0.5M$. Meanwhile,

$$
P(z_i|m=M) = C_M^{z_i} p_i^{w_i}(1-p_i)^{z_i} > 0.
$$

Thus, when $n(p_i > 0.5) \leq n(p_i < 0.5)$,

$$
\lim_{M \to \infty} |\mathbb{E}_{\mathcal{L}}\hat{p}_i - p_i| > 0.
\tag{28}
$$

Combining Equations 26 and 28 concludes Theorem 3. Then, we prove the incentive compatibility of our Bayesian peer prediction mechanism as follows:

Theorem 3. *When the number of tasks is very large, exerting high efforts ($e_i = 1$) and reporting truthfully ($r_i = 1$) are a Bayesian Nash equilibrium for all workers.*

Proof. Suppose all workers except worker i exert high efforts and report truthfully. Since we have made the assumption that $p_{j,H} > 0.5$ for all $j \in [N]$[1], then $n(p_j > 0.5) \geq N - 1 > n(p_j < 0.5)$. In this case, $\mathbb{E}R(i) \approx M \cdot [a \cdot (p_i - 0.5) + b]$. Considering Theorem 1, $\mathbb{E}R(i)$ only reaches the maximum when $e_i = 1$ and $r_i = 1$. Thus, worker i will lose her utility if she does not exert high efforts or report falsely. Thus, ($e_i = 1, r_i = 1$) is a Bayesian Nash equilibrium for all workers. $\square$

---

[1]This assumption requires that if workers work exert high efforts, they should be better than random guess. This is a very weak assumption. If it is violated, then the tasks may not suitable for non-expert crowdsourcing workers.

3.4 Empirical Evaluation

(Empirically Compares with existing ones on stability and fairness. Here, I give an example for the disadvantage of the peer prediction mechanism proposed by [Dasgupta and Ghosh, 2013]. Suppose all workers always report the fully correct answers by exerting high efforts and report truthfully. Then, if the real true labels are all 1, then the payment for workers are 0. If the real true labels are uniformly distributed, the payment are 0.5. Even though the incentive compatibility is ensured by $\geq$, you see, it has some unreasonable parts which, in principle, can be overcome by our mechanism.)

4 Reinforcement Bayesian Peer Prediction

Suppose the true label is 1.

$$
x = \log \frac{P(L=1)}{P(L=2)} = g + \sum_{i=1}^{N} f(x_i, y_i, w_i, z_i)
\tag{29}
$$

where $g = g_1 - g_2$, and

$$
g_1 = \log(s_1 + t_2 + 1) , \ g_2 = \log(s_2 + t_1 + 1).
$$

Omitting the subscript in f, we can have $f = f_1 - f_2$ with probability p and $f = f_2 - f_1$ with probability $1 - p$. Here,

$$
f_1 = \log(x + z + 2) , \ f_2 = \log(w + y + 1).
$$

Thus, we can have

$$
\mathbb{E}g = \mathbb{E}g_1 - \mathbb{E}g_2 , \ \mathbb{E}f = (2p-1)(\mathbb{E}f_1 - \mathbb{E}f_2).
\tag{30}
$$

From the previous proof, we know that $P(m)$ is very small when $m >> 1$. Thus, we mainly focus on the region where m is relatively small. For a given small m,

$$
\mathbb{E}_{s_1, t_2}g_1 \approx \mathbb{E}_{t_2} \log(np + t_2 + 1)
\tag{31}
$$

$$
\log(np + t_2 + 1) = \log(np + 1) + \sum_{i=1}^{\infty}(-1)^{i-1}q^i
\tag{32}
$$

$$
q = \frac{t_2}{np+1} \Rightarrow 0 \leq q^i \leq c^i \cdot \left(\frac{m}{M}\right)^i \leq c^i \frac{m}{M}
\tag{33}
$$

Then,

$$
\mathbb{E}g_1 \approx \mathbb{E}_m \log(1 + Mp - mp)
\tag{34}
$$

$$
\log(1 + Mp - mp) \approx \log(1 + Mp) + \sum_{i=1}^{\infty}(-1)^i \left(\frac{m}{M}\right)^i
\tag{35}
$$

Using the similar way of approximation for the computation of $\mathbb{E}g_2$, $\mathbb{E}f_1$ and $\mathbb{E}f_2$, we can have

$$\begin{aligned}\mathbb{E}g_1 \approx \log(Mp) \ , \ \mathbb{E}g_2 \approx \log(M(1-p))\\ \mathbb{E}f_1 \approx \log(Mp) \ , \ \mathbb{E}f_2 \approx \log(M(1-p))\end{aligned} \qquad (36)$$

Thus, if all workers exert high efforts and report truthfully,

$$\mathbb{E}x \approx \log \lambda_0 + \sum_{i=1}^{N} \log \lambda_{i,H} \qquad (37)$$

If the true label is 2, then

$$\mathbb{E}x \approx \log \lambda_0 - \sum_{i=1}^{N} \log \lambda_{i,H} \qquad (38)$$

Thereby,

$$\mathbb{E}|x| \approx (2p_0 - 1)\log \lambda_0 + \sum_{i=1}^{N} \log \lambda_{i,H} \qquad (39)$$

When, for example, worker 1 deviate from the desired equilibrium strategy, the non-equilibrium state correspond to

$$\mathbb{E}|x'| \approx (2p_0 - 1)\log \lambda_0 + \log \lambda_1 + \sum_{i=2}^{N} \log \lambda_{i,H} \qquad (40)$$

The minimal value of $\mathbb{E}|x'|$ is reached when worker 1 exert high efforts and report falsely, namely $\log \lambda_i = -\log \lambda_{i,H}$. Thus, the maximal reward increment brought by worker 1's strategy switch is

$$V_1 = F(\mathbb{E}x) - F(\mathbb{E}x') \approx 2\log \lambda_{1,H} \cdot \left. \frac{\mathrm{d}F}{\mathrm{d}x} \right|_{x=x_H} \qquad (41)$$

Since p_0 is difficult to estimate, we define the upper bound of the value increment as

$$V = 2\max_i \log \lambda_{i,H} \cdot \max_{x \in [x_H, \infty)} \frac{\mathrm{d}F}{\mathrm{d}x} \qquad (42)$$

where $x_H = \log \lambda_0 + \sum_{i=1}^{N} \log \lambda_{i,H}$. Considering the discounted reward calculation in reinforcement learning, we can know the maximum value difference can be created by the manipulation of any worker is $(1 - \rho)^{-1}V$. Meanwhile, if the reinforcement part increases the scaling factor by δ to obtain the reward increment, we need to pay more than $\sum_{i=1}^{N} M\delta(p_{i,H} - 0.5)$. Thus, if we want to prevent the reinforcement learning module from the adversarial manipulation, the minimal gap δ between to two available scaling factors should satisfy

$$\sum_{i=1}^{N} M\delta(p_{i,H} - 0.5) > V. \qquad (43)$$

References

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

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

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