

---

# Supplementary File for Incentivizing High Quality Crowdsourcing Information using Bayesian Inference and Reinforcement Learning

---

Anonymous Authors<sup>1</sup>

## 1. Introduction

### 1.1. 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.

### 1.2. 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.

## 2. Related Work

## 3. Learning-Based Peer Prediction

### 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  $\mathcal{T} = \{1, 2, \dots, M\}$  and  $\mathcal{C} = \{1, 2, \dots, N\}$ , 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 (?), we assume that  $p_{i,H} > p_{i,L} \geq 0.5$ . Then, we can have Proposition 1.

**Proposition 1.**  $p_i$  is strictly increasing with  $e_i$  and  $r_i$ .

### 3.2. One-Shot 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:

$$R(i) = M \cdot [a \cdot (\hat{p}_i - 0.5) + b] \quad (2)$$

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

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

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

where  $\mathbf{p} = [p_i]_N$  and  $\boldsymbol{\tau} = [\tau_1, \tau_2]$ .  $\tau_1$  and  $\tau_2$  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 (?), we assume a Dirichlet prior for both  $\mathbf{p}_i$  and  $\boldsymbol{\tau}$  as

$$[p_i, 1 - p_i] \sim \text{Dir}(\alpha_1, \alpha_2), \quad \boldsymbol{\tau} \sim \text{Dir}(\beta_1, \beta_2). \quad (4)$$

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

$$\begin{aligned} P(\mathcal{L}, \mathbf{L}, \mathbf{p}, \boldsymbol{\tau} | \boldsymbol{\alpha}, \boldsymbol{\beta}) &= P(\mathcal{L}, \mathbf{L} | \mathbf{p}, \boldsymbol{\tau}) \cdot P(\mathbf{p}, \boldsymbol{\tau} | \boldsymbol{\alpha}, \boldsymbol{\beta}) \\ &= \frac{1}{B(\boldsymbol{\beta})} \prod_{k=1}^K \tau_k^{\hat{\beta}_k - 1} \cdot \prod_{i=1}^N \frac{1}{B(\boldsymbol{\alpha})} p_i^{\hat{\alpha}_{i1} - 1} (1 - p_i)^{\hat{\alpha}_{i2} - 1} \end{aligned} \quad (5)$$

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

$$\begin{aligned} \hat{\alpha}_{i1} &= \sum_{j=1}^M \sum_{k=1}^K \delta_{ijk}^t \xi_{jk}^t + \alpha_1 \\ \hat{\alpha}_{i2} &= \sum_{j=1}^M \sum_{k=1}^K \delta_{ij(3-k)}^t \xi_{jk}^t + \alpha_2 \\ \hat{\beta}_k &= \sum_{j=1}^M \xi_{jk}^t + \beta_k. \end{aligned} \quad (6)$$

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

$$B(x, y) = \frac{(x-1)!(y-1)!}{(x+y-1)!}. \quad (7)$$

Furthermore, we can conduct marginalization via integrating Equation 5 over all possible values of  $\mathbf{p}$  and  $\boldsymbol{\tau}$  as

$$\begin{aligned} P(\mathcal{L}, \mathbf{L} | \boldsymbol{\alpha}, \boldsymbol{\beta}) &= \int_{\mathbf{p}, \boldsymbol{\tau}} P(\mathcal{L}, \mathbf{L}, \mathbf{p}, \boldsymbol{\tau} | \boldsymbol{\alpha}, \boldsymbol{\beta}) d\mathbf{p} d\boldsymbol{\tau} \\ &= \frac{B(\hat{\boldsymbol{\beta}})}{B(\boldsymbol{\beta})} \cdot \prod_{i=1}^N \frac{B(\hat{\boldsymbol{\alpha}}_i)}{B(\boldsymbol{\alpha})} \end{aligned} \quad (8)$$

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

$$P(\mathbf{L} | \mathcal{L}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{P(\mathcal{L}, \mathbf{L} | \boldsymbol{\alpha}, \boldsymbol{\beta})}{P(\mathcal{L} | \boldsymbol{\alpha}, \boldsymbol{\beta})} \propto B(\hat{\boldsymbol{\beta}}) \prod_{i=1}^N B(\hat{\boldsymbol{\alpha}}_i). \quad (9)$$

Note that the previous studies have shown that we should be optimistic about workers' willingness to provide the correct label (?), 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

$$P(L(j) | \mathcal{L}, \mathbf{L}(\bar{j}), \boldsymbol{\alpha}, \boldsymbol{\beta}) \propto P(\mathbf{L} | \mathcal{L}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \quad (10)$$

where  $\mathbf{L}(\bar{j})$  denotes the true labels of all tasks except for task  $j$ . Then, we can generate the samples of the posterior distribution  $P(\mathbf{L} | \mathcal{L}, \boldsymbol{\alpha}, \boldsymbol{\beta})$  by using Algorithm 1. In each round of sampling, Algorithm 1 traverses all tasks by increasing  $j$  from 1 to  $M$  and always update the true label vector  $\mathbf{L}$  via replacing  $L(j)$  with the newly obtained sample (line 3-6). Here, we write the  $s$ -th sample as  $\mathbf{L}^{(s)}$ . 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

$$P(L(j) = k) = \frac{1}{W - b} \sum_{s=b+1}^W \mathbb{1}(L^{(s)}(j) = k). \quad (11)$$

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

$$\hat{L}^t(j) = \arg \max_{k \in \{1, 2\}} \Pr(L^t(j) = k). \quad (12)$$

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

$$\hat{p}_i = \frac{\sum_{s=b+1}^W [\alpha_{i1} + \sum_{j=1}^M \mathbb{1}(L^{(s)}(j) = L_i(j))]}{(W - b) \cdot (\alpha_{i1} + \alpha_{i2} + M)}. \quad (13)$$

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

**Algorithm 1** Gibbs sampling for crowdsourcing

**Input:** the collected labels  $\mathcal{L}$ , the number of samples  $W$   
**Output:** the sample sequence  $\mathcal{S}$   
 $\mathcal{S} \leftarrow \emptyset$ , Initialize  $\mathbf{L} = [L(j)]_M$  with the uniform distribution  
**for**  $s = 1$  **to**  $W$  **do**  
     **for**  $j = 1$  **to**  $M$  **do**  
         Set  $L(j) = 1$  and compute  $x_1 = B(\hat{\beta}) \prod_{i=1}^N B(\hat{\alpha}_i)$   
         Set  $L(j) = 2$  and compute  $x_2 = B(\hat{\beta}) \prod_{i=1}^N B(\hat{\alpha}_i)$   
          $L(j) \leftarrow$  Sample  $\{1, 2\}$  with  $P(1) = x_1/(x_1 + x_2)$   
     **end for**  
     Append  $\mathbf{L}$  to the sample sequence  $\mathcal{S}$   
**end for**

**3.3. Reinforcement Peer Prediction Framework**
**4. Game-Theoretic Analysis**
**4.1. One-Shot Peer Prediction Analysis**

To study the game-theoretical behavior of our mechanism, we firstly define a new function set  $H(m, p; M, t)$  as

$$H(m, p; M, t) = 2^{M+1} \times \int_0^1 [(2p-1)x + 1 - p]^{M-m} [(1-2p)x + p]^m x^t dx \quad (14)$$

where  $M \geq m \geq 0$ ,  $t \geq 0$  and  $1 \geq p \geq 1$ . Besides, for the simplicity of notation, we introduce  $n$  and  $\bar{p}$  as  $n = M - m$  and  $\bar{p} = 1 - p$ . Then, we can have:

**Proposition 2.**  $H(m, 0.5; M, t) = 2(t+1)^{-1}$ .

**Proposition 3.**  $H(m, p; M, t) = H(n, \bar{p}; M, t)$ .

**Proposition 4.** As a function of  $m$ ,  $H(m, p; M, t)$  is logarithmically convex.

*Proof.* Proposition 2 can be proved by integrating  $2x^t$  on  $[0, 1]$ . Proposition 3 can be proved by showing that  $H(n, \bar{p}; M, t)$  has the same expression as  $H(m, p; M, t)$ . Thus, in the following proof, we focus on Proposition 4. Fixing  $p$ ,  $M$  and  $t$ , we denote  $\log(H)$  by  $f(m)$ . Then, we compute the first-order derivative as

$$H(m)f'(m) = 2^{M+1} \int_0^1 \lambda u^n (1-u)^m x^t dx \quad (15)$$

where  $u = (2p-1)x + 1 - p$  and  $\lambda = \log(1-u) - \log(u)$ . Furthermore, we can solve the second-order derivative as

$$2^{-2(M+1)} H^2(m) f''(m) = \int_0^1 g^2(x) dx \int_0^1 h^2(x) dx - \left( \int_0^1 g(x) h(x) dx \right)^2 \quad (16)$$

where the functions  $g, h : (0, 1) \rightarrow \mathbb{R}$  are defined by

$$g = \lambda \sqrt{u^n (1-u)^m}, \quad h = \sqrt{u^n (1-u)^m}. \quad (17)$$

By the Cauchy-Schwarz inequality,

$$\int_0^1 g^2(x) dx \int_0^1 h^2(x) dx \geq \left( \int_0^1 g(x) h(x) dx \right)^2 \quad (18)$$

we can know that  $f''(m) \geq 0$  always holds, which concludes that  $f$  is convex and  $H$  is logarithmically convex.  $\square$

For the special case that  $t = 1$  and  $M \gg 1$ , we can further derive the following three propositions for  $H(m, p; M, 1)$ :

**Proposition 5.** The ratio between two ends satisfies

$$\log \frac{H(0, p; M, 1)}{H(M, p; M, 1)} \approx \begin{cases} \log(M) + \epsilon(p) & p > 0.5 \\ 0 & p = 0.5 \\ -\log(M) - \epsilon(\bar{p}) & p < 0.5 \end{cases}$$

where  $\epsilon(p) = \log(2p-1) - \log(p)$ .

**Proposition 6.** The lower bound can be calculated as

$$\log H(m, p) \gtrsim \begin{cases} H(0, p) - \underline{k}m & 2m \leq M \\ H(M, p) - \underline{k}n & 2m > M \end{cases}$$

where  $\underline{k} = \log(\max\{p/(\bar{p} + M^{-1}), \bar{p}/(p + M^{-1})\})$ .

**Proposition 7.** The upper bound can be calculated as

$$\log H(m, p) \lesssim \begin{cases} H(0, p) - \bar{k}m & 2m \leq M \\ H(M, p) - \bar{k}n & 2m > M \end{cases}$$

where  $n = M - m$  and  $\bar{k} = 2 \log(2 \cdot \max\{p, \bar{p}\})$ .

*Proof.* By Proposition 2,  $\log H(m, 0.5; M, 1) \equiv 0$ , which proves the above three propositions for the case that  $p = 0.5$ . Considering the symmetry ensured by Proposition 3, we thus focus on the case that  $p > 0.5$  in the following proof and transform  $H(m, p)$  into the following formulation

$$H(m, p) = \omega(p) \cdot \int_{\bar{p}}^p x^n (1-x)^m (x-1+p) dx \quad (19)$$

where  $\omega(p) = 2^{M+1}/(2p-1)^2$ . Then, we can solve  $H(0, p)$  and  $H(M, p)$  as

$$\begin{aligned} H(0, p) &= \omega(p) \int_{\bar{p}}^p x^M (x-\bar{p}) dx \\ &= \frac{(2p)^{M+1}}{(2p-1)(M+1)} - O\left(\frac{(2p)^{M+1}}{M^2}\right) \end{aligned} \quad (20)$$

$$\begin{aligned} H(M, p) &= \omega(p) \int_{\bar{p}}^p (1-x)^M (x-\bar{p}) dx \\ &= \frac{p(2p)^{M+1}}{(2p-1)^2(M+1)(M+2)} - O\left(\frac{(2p)^{M+1}}{M+2}\right). \end{aligned} \quad (21)$$

Using the Taylor expansion of function  $\log(x)$ , we can calculate the ratio in Proposition 5 as

$$\log \frac{H(0, p)}{H(M, p)} = \log(M) + \log \frac{2p-1}{p} + O\left(\frac{1}{M}\right) \quad (22)$$

which concludes Proposition 5 when  $M \gg 1$ .

Furthermore, we can solve  $H(1, p)$  as

$$\begin{aligned} H(1, p) &= \omega(p) \int_{\bar{p}}^p x^{M-1} (x - \bar{p}) dx - H(0, p) \\ &= \frac{(2\bar{p} + M^{-1})(2p)^M}{(2p-1)(M+1)} - O\left(\frac{(2p)^{M+1}}{M^2}\right) \end{aligned} \quad (23)$$

The value ratio between  $m = 0$  and  $m = 1$  then satisfies

$$\log \frac{H(1, p)}{H(0, p)} = \log \frac{p}{\bar{p} + M^{-1}} + O\left(\frac{1}{M}\right). \quad (24)$$

By Rolle's theorem, there exists a  $c \in [m, m+1]$  satisfying

$$\log H(1, p) - \log H(0, p) = f'(c) \quad (25)$$

where  $f(m) = \log H(m, p)$ . Meanwhile, Proposition 4 ensures that  $f''(m) \geq 0$  always holds. Thus, we can have

$$\log H(m+1, p) - \log H(m, p) \geq \log \frac{H(1, 0)}{H(0, p)} \quad (26)$$

which concludes the first case of Proposition 6. Similarly, we compute the ratio between  $m = M-1$  and  $M$  as

$$\log \frac{H(M, p)}{H(M-1, p)} = \log \frac{p}{\bar{p} + M^{-1}} + O\left(\frac{1}{M}\right). \quad (27)$$

Meanwhile, Rolle's theorem and Proposition 4 ensure that

$$\log H(m, p) - \log H(m-1, p) \leq \log \frac{H(M, 0)}{H(M-1, p)} \quad (28)$$

which concludes the second case of Proposition 6.

Lastly, we focus on the upper bound described by Proposition 7. According to the inequality of arithmetic and geometric means,  $x(1-x) \leq 2^{-2}$  holds for any  $x \in [0, 1]$ . Thus, when  $2m \leq M$  (i.e.  $n \geq m$ ), we can have

$$H(m, p) \leq 2^{-2m} \omega(p) \cdot \int_{\bar{p}}^p x^{n-m} (x - 1 + p) dx \quad (29)$$

where the equality only holds when  $m = 0$ .

$$\int_{\bar{p}}^p x^{n-m} (x - 1 + p) dx = \frac{(2p-1)p^\delta}{\delta} + \frac{\Delta}{\delta(\delta+1)} \quad (30)$$

where  $\delta = n - m + 1$  and  $\Delta = \bar{p}^{\delta+1} - p^{\delta+1} < 0$ . Hence,

$$\log \frac{H(m, p)}{H(0, p)} \leq -2m[\log(2p) - \varepsilon(m)] + O\left(\frac{1}{M}\right) \quad (31)$$

where  $\varepsilon(m) = -(2m)^{-1}[\log(n-m+1) - \log(M+1)]$ . Since  $\log(x)$  is a concave function, we can know that

$$\varepsilon(m) \leq (M)^{-1} \log(M+1) = O(M^{-1}) \quad (32)$$

which concludes the first case in Proposition 7. Similarly, for  $2m > M$  (i.e.  $n < m$ ), we can have

$$\log \frac{H(m, p)}{H(M, p)} \leq -2n[\log(2p) - \hat{\varepsilon}(n)] + O\left(\frac{1}{M}\right) \quad (33)$$

where  $\hat{\varepsilon}(n) \leq O(M^{-1})$ . Thereby, we can conclude the second case of Proposition 7. Note that the case where  $p < 0.5$  can be derived by using Proposition 3.  $\square$

For the case that  $t = 0$  and  $M \gg 1$ , using the same method as the above proof, we can derive the same lower and upper bounds as Propositions 7 and 6. On the other hand, for  $t = 0$ , Proposition 5 does not hold and we can have

**Proposition 8.**  $H(m, p; M, 0) = H(n, p; M, 0)$

*Proof.* When  $t = 0$ ,

$$H(m, p) = 2^{M+1} (2p-1)^{-1} \int_{\bar{p}}^p x^n (1-x)^m dx. \quad (34)$$

Then, substituting  $x$  as  $1-v$  concludes Proposition 8.  $\square$

Next, we derive the upper bound of the estimation errors for our Bayesian inference algorithm.

**Proposition 9.** 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,

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

*Proof.* For the posterior label vector  $\mathbf{L}$  generated by the posterior distribution  $P(\mathbf{L}|\mathcal{L}, \alpha, \beta)$ , we introduce  $n$  and  $m$  to denote the number of correct and wrong labels, respectively. Then, among the  $n$  tasks of which the posterior label is correct, we introduce  $x_0$  and  $y_0$  to denote the number of tasks of which the real true label is 1 and 2, respectively. Among these  $n$  tasks, we further use  $x_i$  and  $y_i$  to denote the number of tasks of which the label provided by worker  $i$  is correct and wrong, respectively. Among the  $m$  tasks of which the posterior label is wrong, we introduce  $w_0$  and  $z_0$  to denote the number of tasks of which the real true label is 1 and 2, respectively. Among these  $m$  tasks, we also introduce  $w_i$  and  $z_i$  to denote the number of tasks of which the label provided by worker  $i$  is correct and wrong, respectively. Thus, we can have  $n+m = M$ ,  $x_i + y_i = n$ ,  $w_i + z_i = m$  and  $\mathbb{E}_{\mathcal{L}, \mathbf{L}}(x_i + w_i) = Mp_i$ . Meanwhile, in Equation 13, when  $M \gg 1$ , we have  $M\hat{p} \approx \mathbb{E}_{\mathbf{L}}(x_i + z_i)$ . Thus, the inference error can be calculated as

$$|\mathbb{E}_{\mathcal{L}, \mathbf{L}} \hat{p}_i - p_i| \approx \mathbb{E}_{\mathcal{L}, \mathbf{L}} \frac{|w_i - z_i|}{M} \leq \mathbb{E}_{\mathcal{L}, \mathbf{L}} \left[ \frac{m}{M} \right]. \quad (35)$$

To compute the expectation, we need to analyze the probability density function of  $m$ . According to Equation 9, we can know the probability density function  $P(m)$  satisfies

$$P(m) = \frac{C_M^m}{Z} F(m, p_0; M, 0) \prod_{i=1}^N F(m, p_i; M, 1) \quad (36)$$

where  $Z$  is the normalization constant and

$$F(m, p; M, t) = \sum_{x=0}^n \sum_{w=0}^m 2^{M+1} C_n^x C_m^w \times p^{x+w} (1-p)^{y+z} B(x+z+1+t, y+w+1). \quad (37)$$

Here,  $n = M - m$ . Note that, when deriving Equation 37, we utilize the independence between different workers. Besides, in Equation 37, we add the factor  $2^{M+1}$  to facilitate building the connection between  $F$  and the  $H$  function defined in Equation 14. Due to the normalization constant  $Z$ , the factor  $2^{M+1}$  will not affect the value of  $P(m)$ .  $\square$