Reward or Penalty: Aligning Incentives of Stakeholders in Crowdsourcing

Jinliang Xu, Shangguang Wang, *Senior Member, IEEE,* Ning Zhang, *Member, IEEE*, Fangchun Yang, *Senior Member, IEEE*, and Xuemin (Sherman) Shen, *Fellow, IEEE*

Abstract—Crowdsourcing is a promising platform, whereby massive tasks are broadcasted to a crowd of semi-skilled workers by the requester for reliable solutions. In this paper, we consider four key evaluation indices of a crowdsourcing community (i.e. quality, cost, latency, and platform improvement), and demonstrate that these indices involve the interests of the three stakeholders, namely requester, worker and crowdsourcing platform. Since the incentives among these three stakeholders always conflict with each other, to elevate the long-term development of the crowdsourcing community, we take the perspective of the whole crowdsourcing community, and design a crowdsourcing mechanism to align incentives of stakeholders together. Specifically, we give workers reward or penalty according to their reporting solutions instead of only nonnegative payment. Furthermore, we find a series of proper reward-penalty function pairs and compute workers personal order values, which can provide different amounts of reward and penalty according to both the workers reporting beliefs and their individual history performances, and keep the incentive of workers at the same time. The proposed mechanism can help latency control, promote quality and platform evolution of crowdsourcing community, and improve the aforementioned four key evaluation indices. Theoretical analysis and experimental results are provided to validate and evaluate the proposed mechanism respectively.

Index Terms—Crowdsourcing, incentive, reward, penalty, belief.

1 Introduction

The common application of crowdsourcing is in the contexts of knowledge gathering (e.g. mobile crowdsensing[1]) or decision making tasks (labeling of training dataset in machine learning). In these contexts, the number of tasks to complete is too large for insufficient number of experts, and the evaluation process cannot be automatically performed very well by a computer [2], [3], [4]. As a result, a feasible alternative is to resort to a crowd of individuals (i.e. *workers*) recruited on an online *crowdsourcing platform* (e.g., MTurk[1] or CrowdFlower[2]) to undertake these tasks (i.e. completing a task, then reporting the solution of the task through the crowdsourcing platform) [5]. The person who publishes tasks and obtains the solutions through the crowdsourcing platform is called the *requester*.

Based on the state of art of crowdsourcing industry and academia, we summarize four key evaluation indices of current crowdsourcing community, namely *quality*, *cost*, *latency* and *platform improvement*: 1) **Quality**. In typical crowdsourcing settings, like MTurk and CrowdFlower, a worker is simply paid in proportion to the amount of tasks she has completed. As a result, a worker inclines to undertake tasks that she is not good at, or spends less effort and time on each task, thereby degrading the quality of her reportings [6]. However, the requester desires workers to report high-quality solutions, as a task's final truthful solution is elicited from the collected solutions; 2) **Cost**. The cost control focuses on how to motivate the workers to do their best with minimal cost [7], [6]. It is reasonable to assume that both the requester and workers are self-interested and rational [8], [9], [10], [11]. Hence each worker attempts to maximize her own payment, while the requester aims to achieve high-quality final solutions of tasks with minimal cost; 3) **Latency**. Latency control is important as the practical total completion time for the whole tasks of a requester may exceed the time constraint set by the requester [12]. Excessive latency may occur when professional workers are insufficient, or tasks are difficult for most average work-

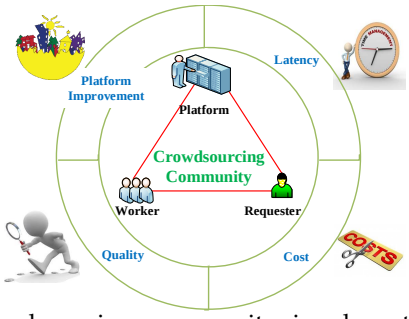


Fig. 1: Crowdsourcing community involves three stakeholders, namely *requester*, *worker* and *crowdsourcing platform*. They interact with each other through the four key evaluation indices of the whole crowdsourcing community, namely *quality*, *cost*, *latency* and *platform improvement*.

- *Jinliang Xu, Shangguang Wang, and Fangchun Yang are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. E-mail: jlxu@bupt.edu.cn; sgwang@bupt.edu.cn; fcyang@bupt.edu.cn,*
- *Ning Zhang and Xuemin(Sherman) Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, N2L 3G1. E-mail: n35zhang@uwaterloo.ca; sshen@uwaterloo.ca,*

xxxx-xxxx/0x/$xx.00 2017 IEEE Published by the IEEE Computer Society

1. http://www.mturk.com/mturk/welcome
2. https://www.crowdflower.com/

ers. In these cases, if a difficult task is skipped by most workers, it will lag behind and result in excessive latency phenomenon of the whole tasks; 4) and **Platform Improvement**. Platform improvement targets on the sustainable development of the crowdsourcing community. This includes attracting more professional or reliable workers, meanwhile preventing badly-behaved workers (i.e., the opposite of a professional workers, whose reportings in history are typically judged wrong) flooding into the crowdsourcing platform. Important though, platform improvement has not been explicitly put forward and studied currently [7].

From the above description of the four key evaluation indices, we demonstrate that crowdsourcing community involves the interests of three stakeholders, namely *requester*, *worker* and *crowdsourcing platform* (Fig. 1). Existing crowdsourcing community seldom considers aligning incentives of the stakeholders together, which can be a significant obstacle to its further application [13]. Moreover, the incentives among the three stakeholders always conflict with each other. This has an adverse effect on the four key evaluation indices [14], [15], [16], [17], which will impede the long-term development of the whole of a crowdsourcing community (mainly referring to the three stakeholders). For example, if a worker aims to earn more payment (the fundamental objective of workers), she may attempt to complete as many tasks as possible within fixed time cost, thereby resulting in low quality of the reportings; if she also desires to gain high *approval rate* (the percentage of reporting solutions that are judged right, which is an important indicator adopted by MTurk and CrowdFlower), she may skip the difficult tasks (*skipping* is a function supported by most crowdsourcing platform), thereby leading to excessive latency. Neither of the worker's strategies are in line with the requester's incentive. If workers and requesters cannot reach a consensus, they may leave the crowdsourcing community for job-hopping, which will significantly harm the crowdsourcing platform. Hence, the key to improving all of the four key evaluation indices is to take the perspective of the whole crowdsourcing community and design a mechanism that can align together the incentives of three stakeholders in the crowdsourcing community.

To this end, in this paper, we design a mechanism to align the incentives of three stakeholders in current crowdsourcing. More specifically, the major contributions include:

1) We show the conflict of interest of three stakeholders in crowdsourcing community. Further, we point out the importance of aligning their incentives for improving system performance.
2) We conduct a questionnaire survey among 500 workers on CrowdFlower, and verify that our hypothesis that in most cases in real crowdsourcing community all workers believe that they observe the real solution of each task, which is only perturbed by unbiased noise.
3) We find a series of proper *reward-penalty function pairs* and design a mechanism to align together the incentives of the three stakeholders. Compared with a single reward function or reward-penalty function pair, the proposed mechanism is more effective to control the cost of the requesters, keep the incentive of workers, and improve the performance of platform.
4) Both theoretical analysis and simulation experiments demonstrate that the proposed mechanism can help to improve the four key evaluation indices of crowdsourcing.

The remainder of this paper is organized as follows: we first present and verify a hypothesis in crowdsourcing in section 2. Then, we propose the incentive mechanism to align the incentives of different parties in crowdsourcing in section 3. In section 4, we theoretically analyze and study the properties of the proposed mechanism. In Section 5, we implement the proposed mechanism on existing crowdsourcing platform and conduct simulation experiments to further validate the proposed mechanism. Section 6 presents the existing literature and Section 7 closes the paper with conclusions.

2 Verification of The Hypothesis

The proposed mechanism is mainly based on the following hypothesis: all workers believe that in most cases they observe the real solution of each task, which is only perturbed by unbiased noise. An example to support this assumption is when some workers in the same classroom are asked to count the number of students, and they make decisions on their own and are not allowed to communicate with each other. A radically different assumption from the mentioned hypothesis is that a worker can obtain the intention or preference of other workers. For instance, workers are asked to report their attitudes towards a well-known social issue (e.g. public voting), the statistical information of people's attitudes can be obtained from the media or other ways, and a worker will report what others will report to get a payment, regardless of the real solution.

In order to verify our hypothesis, we conduct a questionnaire survey among 500 workers on CrowdFlower. In this survey, we ask the workers just to report the percentage of cases where they need to take into account the attitudes of other workers on CrowdFlower when performing the crowdsourcing tasks, i.e. the probability that our hypothesis holds in real crowdsourcing community. Based on the results of the questionnaire survey, we show the probability density estimation curve [18] in Fig. 2. The probability density estimation curve in Fig. 2 is highly asymmetry, and it shows that our hypothesis holds in most cases in a real crowdsourcing community. Specifically, nearly 95% of workers think the probability that our hypothesis holds is large than 0.6, while more than 80% of workers think the probability is large than 0.8. Note that the hypothesis is not necessarily true for all the settings, however, it holds in most cases in a real crowdsourcing community. We continue to conduct a questionnaire survey among 400 workers on another crowdsourcing community named MTurk, which is a famous crowdsourcing community like CrowdFlower. The result shows that nearly 92% of workers think the probability that our hypothesis holds is large than 0.6, while more than 84% of workers think the probability is large than 0.8.

Though the proposed mechanism can be applied in real crowdsourcing community such as CrowdFlower, it should be noted that it is designed mainly for the situations where the hypothesis holds. While in dealing with social problems such as a public opinion poll, or social problems take up the

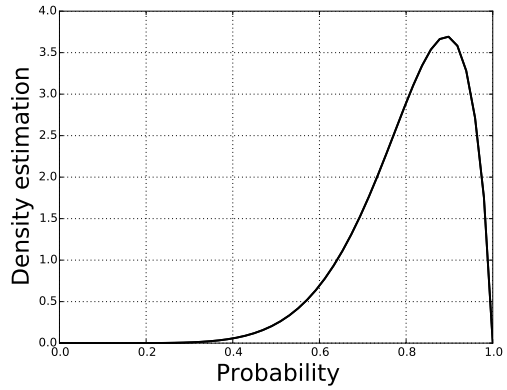


Fig. 2: The probability density estimation curve of the probability that our hypothesis holds in real crowdsourcing community.

major part, it will be not appropriate. In fact, several existing theoretical development of crowdsourcing mechanisms have partially or totally work on this hypothesis too, such as Peer Prediction [15], and Bayesian Truth Serum[16], and the deployment of simple and intuitive mechanisms, such as the Output Agreement mechanism [19]. Different from these works, the proposed mechanism may reward or punish a worker, instead of giving only nonnegative payment, which will be described in detail later. What is more, we further analyze and divided the possible cases into two kinds of hypothesizes, and definitely show that we are working on the former.

3 Incentive Mechanism Design

The basic idea of the proposed mechanism is as follows. A worker is required to report both of *type* (i.e. her selected solution for a task[3].) and *belief* values (i.e. her confidence of the selection, or the probability that her solution for the task is judged right) for a task. The reason for introducing belief is that people always estimate different probabilities of giving the right answer intentionally or unintentionally when faced with a decision making problem, due to the inherent difficulty of problem, their different experiences and professional knowledge, etc. [20]. With the reporting tuples <type, belief> of several workers for the same task, a mechanism is devised to utilize them to generate the task's *benchmark solution* and its *final truthful solution*. If a worker's reporting type is the same as the benchmark solution, she will get a reward from the requester, otherwise she must pay a penalty back to the requester. The rationale is that the former has a positive influence on generating the task's benchmark solution, while the latter is the opposite. In addition, for a reporting tuple, the amount of reward and penalty has a positive correlation with the worker's reporting belief as the influence of reporting tuples increases with belief value [20], [21]. We should find the proper reward-penalty function, with which a worker can maximize the expected payment if and only if she truthfully reports the type and the associated belief. Note that the final truthful solution is different from the benchmark solution, which will be detailed later.

3. The candidate solutions for a task is finite, and every candidate solution is a potential reporting type.

The proposed mechanism is composed of three rules, namely *judgement rule*, *reward and penalty rule*, and *final generating rule*. As shown in Fig. 3, the judgement rule takes reporting types as input to judge whether a worker deserves reward or penalty, instead of paying only nonnegative money in most existing studies; the reward and penalty rule decides the amount of reward and penalty according to the reporting beliefs; and the final generating rule generates the final solution for a task, based on both the reporting types and beliefs. The combination of *judgement rule* and *reward and penalty rule* stimulates workers to truthfully reports their types and beliefs for the given tasks. And the *final generating rule* helps to generate reliable final solutions. In the following, we will detail the three rules step by step.

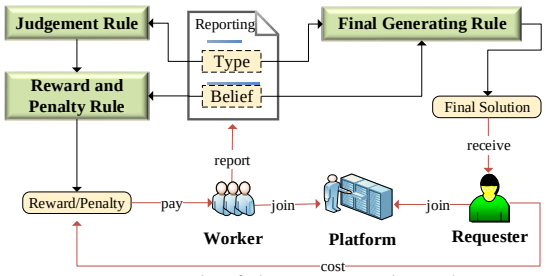


Fig. 3: Framework of the proposed mechanism.

3.1 Judgement Rule

It is worth noting that in this work we only focus on binary-type tasks (the candidate solutions for a task is binary, e.g. *Yes/No*, *Right/Wrong* or *A/B*). This is mainly due to the two reasons: On the one hand, we observe that a number of interesting judgement tasks (e.g. adult image identification) are indeed binary [22], [16]. On the other hand, a task with more than two candidate solutions can be readily broken up into several binary-type tasks [5]. For example, eliciting people attitudes from three options, i.e. positive, negative or neutral on the British voting to leave the European Union in 2016[4] is a three-type task. We can decompose it into three binary-type tasks, i.e. whether a person's attitude is positive (negative or neutral) or not.

Suppose that a requester has published T tasks on a crowdsourcing platform, and there are totally N workers that have participated in undertaking these tasks. The task is indexed by $t \in \{1, 2, \cdots, T\}$, and the worker is indexed by $n \in \{1, 2, \cdots, N\}$. For convenience, we denote the binary-type space as $\mathcal{Y} = \{-1, +1\}$. For example, in a *Yes/No* questionnaire problem, we can denote Yes with $+1$ and No with -1. Let s_t denote the set of workers that have completed task t, such that $s_t \subset \{1, 2, \cdots, N\}$. If worker n has completed task t, we let $y_{n,t} \in \mathcal{Y}$ denote her reporting type of task t. Then, we have the *benchmark solution* of task t, which is defined as follows:

$$\widehat{y_t} = \text{sign}\left(\sum_{n \in s_t} y_{n,t}\right), \tag{1}$$

where function $\text{sign}(\pi) = 1$ if $\pi \geq 0$ and -1 otherwise. If $\sum_{n \in s_t} y_{n,t} = 0$, Eq. 1 will not generate a valid benchmark

4. https://en.wikipedia.org/wiki/United_Kingdom_European_Union_membership_referendum,_2016

solution. To deal with this problem, we can add another worker to complete the task, and compute Eq. 1 again. Note that the reporting belief associated to the reporting type is not considered into Eq. 1, because a simple majority voting rule can avoid the inference of worker's reporting beliefs in generating the benchmark, which is used to judge whether a worker deserves reward or penalty.

Benchmark solution $\widehat{y}_t$ is used to judge whether a worker deserves reward or penalty for her reporting type $y_{n,t}$ of task t. Specifically, for a worker $n \in s_t$, if $y_{n,t}$ is equal to $\widehat{y}_t$, she will get a reward from the requester, otherwise she must pay a penalty back to the requester. Therefore, the judgement rule can be summarized as follows:

$$judgement = \begin{cases} \text{reward}, & \text{if} \quad y_{n,t} = \widehat{y}_t \\ \text{penalty}, & \text{if} \quad y_{n,t} \neq \widehat{y}_t \end{cases}. \tag{2}$$

To resist a collusion attack [23], we can distribute the tasks randomly to a vast number of workers, which will make a collusion attack very expensive to launch successfully.

It is possible to incorporate penalty into the practical crowdsourcing incentive mechanisms. The reasons are 1) A crowdsourcing platform will not always attempt just to attract as many workers as possible. For example, to improve quality, MTurk decided not to accept new international worker accounts in 2012, though this would lose a lot of international workers[5]. 2) Work in [20] shows that worker's confidence for a wrong answer should be imposed a penalty that is related to the confidence value. So forcing workers to pay for their poor performance can be an option. However, the current crowdsourcing platforms such as MTurk, have not adopted penalty as part of incentive mechanism.

Note that the formulation of benchmark solution in Eq. 1 is essentially an application of the *simple majority rule* [24], where the candidate type with more workers is selected as benchmark (with some way of breaking ties, for example, flipping a coin). Therefore, this rule naturally inherits the merits of the simple majority rule, making truthful reporting as a dominant strategy of workers if they want to get more reward.

3.2 Reward and Penalty Rule

In this section, we present the process of finding the proper *reward-penalty function pair* (i.e. the reward function and its corresponding penalty function), with which a worker can maximize the expected payment if and only if she truthfully reports <type,belief> for a task.

The formulation of benchmark solution in Eq. 1 can be transformed as follows:

$$\widehat{y}_t = \arg\max_{\mathcal{Y}} \left\{ \frac{\sum_{n \in s_t, y_{k,n}=+1} 1}{|s_t|}, \frac{\sum_{n \in s_t, y_{k,n}=-1} 1}{|s_t|} \right\}, \tag{3}$$

where $|s_t|$ means the *number of workers for task* t. Note that if $|s_t|$ is large enough, $\frac{\sum_{n \in s_t, y_{k,n}=+1} 1}{|s_t|}$ and $\frac{\sum_{n \in s_t, y_{k,n}=-1} 1}{|s_t|}$ will be very close to the probabilities that workers report type $+1$ and type -1 respectively for task t. As we only consider binary-type tasks, the type with probability larger than 0.5 is surely selected as benchmark. In fact, when a person makes a decision, she always unconsciously estimates the probability that her choice will be proved right. Taking stock market as a similar example [25], investors decide to buy or sell stocks according to personal estimation of price trends (up or down), and in turns the future price is influenced by investors' current decisions.

The concept of reporting belief in this work derives just from the above probability. It is of importance because it indicates the reliability of its corresponding type, which is helpful in generating final truthful solutions. The above probability can be considered as a worker's belief about what other workers will report. While in the application scenario under consideration, we assume every worker has little knowledge of others and she can only depend on her unique experience and knowledge to estimate the answer and her belief, which is a stronger assumption than work in [16]. In addition, the belief value in this work is not shared by workers, and each agent's reporting belief will not update in the process. So a worker's belief about what other workers have reported (the above probability) does not contradict with her confidence of the selection (reporting belief).

The value range of reporting *belief* is set as $[0.5, 1]$, which is different from [16]. In [16], the reporting belief lies in range $[0, 1]$, which supposes belief in $(0.5, 1]$ and $[0, 0.5)$ determinatively correspond to type=1 and 0, respectively. The proposed reporting tuple is not just a different user interface compared to a single report of belief value in range $[0, 1]$. The advantages are as follows: 1) If the belief value in a reporting tuple is in range $[0, 0.5)$, the reporting tuple will be considered invalid for this palpable mistake. As a result, compared to the single reporting belief in range $[0, 1]$ in [16], the proposed reporting tuple can filter out reporting tuples with palpable mistakes; and 2) It aligns agents' reporting belief in the same range [0.5, 1], which makes it easy for us to deduce the family of reward-penalty functions in Eq. 10. The single reporting belief in range $[0, 1]$ in [16] cannot provide us with this convenience.

Suppose a worker gives a reporting belief value x, instead of the real probability of her reporting type in her mind (denoted by symbol c). We let $r(x) > 0$ be the reward function of variable x when worker's reporting type is right, and $p(x) > 0$ be the penalty function of x when worker's reporting type is wrong. The worker's reporting type will be judged right with probability c and wrong with probability $1-c$. Then, the expected gain function of the worker for this reporting is defined as follows:

$$g(x) = r(x) \cdot c - p(x) \cdot (1 - c). \tag{4}$$

Note that the value c of different workers may be different based on their different experiences and professional knowledge. Moreover, the same worker's value c for different tasks may be different due to the inherent difficulty of different tasks. For a specific pair of worker and task, c should be a known value to the worker. While when the task is given to different workers, c may have different values because everyone has unique knowledge and experience from others. Here we consider it in a more holistic perspective instead of the view of an individual worker. That is why we can call it a variable instead of a constant value here.

5. http://turkrequesters.blogspot.sg/2013/01/the-reasons-why-amazon-mechanical-turk.html

Another fact that we should not neglect is that Eq. 2 and Eq. 4 do not conflict with each other. According to Eq. 2, whether a worker gets reward or penalty is based on her reported value y and the values reported by the majority. And the amount of reward and penalty to a worker is decided by her reporting belief value, which is reflected in Eq. 10. As we can see, Eq. 2 and 4 work well in a specific task. In Eq. 4, it says the fact that the expected gain value of a worker in completing massive similar tasks (Or the reporting of the same task by the worker happens many times) is determined by the worker's individual ability, which is relative to her knowledge and experience and the task itself. A complete version of Eq. 4 is Eq. 11. The c in Eq. 11 means the worker's individual ability. So Eq. 2 can be considered as a specific case for a worker and a task, while Eq. 4 describes the statistical result.

To find the proper reward-penalty function pair, namely expressions of $r(x)$ and $p(x)$, we list the following three properties they must satisfy:

1) **Incentive Compatible Principle.** A mechanism is incentive compatible when a worker can maximize the expected payment if and only if she tells the truth [6]. It is reasonable because each worker is self-interested in actual situations. She would not report the value c, if doing so would not bring her benefits or even worse. Under this mechanism, a worker needs not to perform any complex computations to know what value to report. Provided that the worker trusts that the crowdsourcing community operates as prescribed, she will prefer to report honestly [23], [21]. Our goal is to design reward-penalty functions $r(x)$ and $p(x)$ appropriately, to meet the constraint as follows:

$$\arg\max_x g(x) = c, \tag{5}$$

which means that the value of reporting belief x that can maximize $g(x)$ is equal to the real chance c, or c is the maximum value point. Eq. 5 means the incentive compatible principle.

At the position of the maximum value point, the first derivative of a twice differentiable function must be equal to 0, and the second derivative must be negative. So we have

$$\begin{aligned} \frac{r'(x)}{p'(x)} &= \frac{1}{x} - 1 \\ r''(x)\cdot x - p''(x)\cdot(1-x) &< 0 \end{aligned}. \tag{6}$$

2) **Gradients of Reward and Penalty Functions.** The larger of reporting belief, the larger of the influence of its corresponding reporting type. If a worker's reporting type is judged as right, we should reward the worker, and the reward amount should be positively related to her reporting belief x. On the contrary, if the reporting type is judged as wrong, we should punish the worker, and the penalty amount should also be positively related to her reporting belief x. As a result, the gradients of reward-penalty functions must be positive as follows:

$$r'(x) > 0, p'(x) > 0. \tag{7}$$

This basically reflects the fact that the amount of reward and penalty has a positive correlation with the worker's reporting belief since the influence of reporting tuples increases with belief value increases[20], [21].

3) **Bounds of Reward and Penalty Functions.** Boundary values for the differential equations are needed for *easy solvability* [26]. We list two boundary constraints: a) For binary-type tasks, a reporting type with reporting belief $x = 0.5$ provides no useful information, so the reward and penalty function at $x = 0.5$ should both be 0. b) In addition, for simplicity, we set the reward at $x = 1$ with 1, which means that a worker giving a right answer with reporting belief 1 will get one unit reward. Then, we have the bounds of functions of rewards and penalties by the following:

$$r(0.5) = p(0.5) = 0, \quad r(1) = 1. \tag{8}$$

By combining the above constraints together, we can build the mathematical model of the *reward-penalty function pair* as follows:

$$\begin{cases} \frac{r'(x)}{p'(x)} = \frac{1}{x} - 1 \\ r''(x)\cdot x - p''(x)\cdot(1-x) < 0 \\ r'(x) > 0, \qquad r(x) \geq 0 \\ p'(x) > 0, \qquad p(x) \geq 0 \\ r(0.5) = p(0.5) = 0, \quad r(1) = 1 \end{cases}, \tag{9}$$

where $0.5 \leq x \leq 1$. If a function pair of $r(x)$ and $p(x)$ satisfies these constraints above, this function pair is considered to be proper for the proposed crowdsourcing mechanism.

Note that there exist numerous function pairs of $r(x)$ and $p(x)$ without more constraints. For convenience, we only consider the simplest case, where $r(x)$ and $p(x)$ are both polynomial functions of x, and the *polynomial greatest common divisor* of their first derivatives is a power function of x (e.g. x, x^2, etc.). Then, by solving Eq. 9, we can get a family of reward-penalty functions as follows:

$$\begin{cases} r_k(x) = -\frac{(k-1)2^k}{2^k-k-1}x^k + \frac{2^k k}{2^k-k-1}x^{k-1} - \frac{k+1}{2^k-k-1} \\ p_k(x) = \frac{(k-1)2^k}{2^k-k-1}x^k - \frac{k-1}{2^k-k-1} \end{cases}, \tag{10}$$

where $k \geq 2$ is the order of reward-penalty function pairs.

As is expected, if we replace $r(x)$ and $p(x)$ in Eq. 4 with Eq. 10, we can easily find that $x = c$ is the global maximum point of $g(x)$ in range $x \in [0.5, 1]$. As a result, we can write down the k-th order expected gain function by substituting Eq. 10 into Eq. 4, as follows:

$$g_k(c) = \frac{-2\,ck + 2^k c^k + k - 1}{2^k - k - 1}. \tag{11}$$

Note that it is very hard for a worker to provide the exact value of belief for the answer she offers. Note that the expected gain function is a concave function in range [0.5,1] and has only one peak at the exact value. If the belief value the worker provides is closer to the real value, she will gain more. As a result, providing the exact belief value is best, but not necessary.

Another fact that should not be neglected is that every task has a belief value behind it, while workers may give different estimated belief values. The belief value behind each task can be considered as the probability that this task will be given a right solution. We name it as real belief value c. However, the real belief value cannot be obtained by the

crowdsourcing platform and every single workers. To the worker n who tries to complete task t, she can only estimate the a estimated belief value $c_{n,t}$ of the real belief value c according to his knowledge, efforts, and other factors. In most cases, the estimated belief value $c_{n,t}$ is near the real belief value c. To maximize the expected gain, the worker should report truthfully the her solution and estimated belief value $c_{n,t}$. In the latter part, without loss of generality, we still use symbol c to represent $c_{n,t}$ for simplicity.

3.3 Final Generating Rule

The concept of benchmark solution in *judgement rule* considers workers' reporting beliefs are the same, which is not consistent with the actual situation. Therefore, the benchmark solution cannot be considered as the task's final truthful solution. In fact, a reporting type with a larger reporting belief value is more reliable than a smaller one. So a good final generating rule should take this belief value into account.

In what follows, we present the proposed final generating rule. If worker n has completed task t, let $g_{n,t}$ denote the expected payment, and $y_{n,t} \in \mathcal{Y}$ denote the corresponding reporting type. The final solution y_t^* for task t can be computed as follows:

$$y_t^* = \text{sign}\left(\sum_{n \in s_t} g_{n,t} y_{n,t}\right). \quad (12)$$

Compared to the simple majority rule in *judgement rule* (Eq. 1), Eq. 12 is similar to weighted majority rule [24] or weighting aggregation procedure [27], where the weight of reporting type $y_{n,t}$ is obviously the expected payment $g_{n,t}$. The proposed final generating rule can find truth even if most workers' reporting types are wrong.

The expected payment $g_{n,t}$ is chosen as the weight of reporting type $y_{n,t}$ for two reasons: 1) $g_k(c)$ is a monotonically increasing function of variable c in range $[0.5, 1]$ for all $k \geq 2$, which ensures a larger weight if a reporting type has a larger reporting belief value, and vice versa; 2) by incorporating $r_k(x)$ and $p_k(x)$, $g_k(c)$ is more suitable than each of them to measure the reliability of a reporting type; 3) compared to a worker' belief value c, $g_k(c)$ has more distinguishing ability with the quality of workers' reportings.

In this paper, two kinds of solutions are generated, i.e., benchmark solution and final generating solution. The benchmark solution is generated by the simple majority rule to determine whether to reward or punish the workers. Together with the reporting belief value of each reporting solution from a worker, the benchmark solution can even determine the amount of award or penalty. The simple majority rule naturally inherits the merits of the simple majority rule, making truthful reporting as a dominant strategy of workers if they want to get more reward. However, the benchmark solution considers workers' reporting solutions as equal regardless of the different reporting belief values, which is not very impertinent. As a large belief value means a more reliable reporting solution, we must take into account it when generating the final generating solution. As the respected gain function has a lot of good merits, we take it as the weight of reporting solutions in weighted majority rule, and use the result as the final generating rule. In conclusion, the benchmark solution and final generating solution have quite distinct functions in the proposed mechanism, and no one of them is indispensable.

3.3.1 Utility of Reportings

First we define the idea of *utility of reportings*. Then we show that under the proposed mechanism, for a requester, hiring badly-behaved workers does not necessarily incur more costs than hiring professional workers.

Definition 1 (Utility of reportings)**.** A reporting's utility means to what extent it influences the process of generating the final solution for a task. Applying this definition into Eq. 12, we can conclude that in this work the utility of a reporting is the expected payment $g_{n,t}$.

The above definition of *utility of reportings* leads to an intriguing phenomenon in estimating the final truthful solution of a task. As Eq. 12 can be written as follows:

$$y_t^* = \text{sign}\left(+\sum_{n \in s_t, y_{k,n}=+1} g_{n,t} - \sum_{n \in s_t, y_{k,n}=-1} g_{n,t}\right), \quad (13)$$

it shows that adding up individual utilities of the same reporting type (+1 or -1) together can lead to a bigger utility value. Further, if the sum of utilities of several badly-behaved workers (each with a small utility) is equal to a small number of professional workers (each with a bigger utility), the requester will pay them the same. In other words, hiring badly-behaved workers does not necessarily mean to cost more than hiring professional workers.

This above conclusion is quite different from earlier works [28], [8], [15], [16], [19] or current crowdsourcing communities. The fundamental reason is that in these works once a worker's reporting type is judged as right, the requester should pay her the same amount, regardless of the reliability difference of the reporting types. What's more, a requester always gives workers nonnegative payment, instead of giving a penalty to workers with a wrong reporting type. As a result, to get more reliable answers under the cost limitation, the requester always tries as much as possible to select professional workers to complete tasks and avoid badly-behaved workers meanwhile.

To force workers to pay penalty to the requester, a real platform can charge each worker a suitable amount of refundable deposit in advance. If a worker refuses to give the deposit, she will not be allowed to perform any crowdsourcing tasks. For every wrong reporting type, the platform will take a certain amount of money from the worker's deposit, and give it to the requester. If the balance is zero or below a fixed level, the platform is responsible to charge additional deposit from the worker. The advance deposit can avoid the case where the worker is unwilling to pay penalty to the requester for her wrong reporting type. The deposit is refundable. That is to say, when a worker deletes her account someday, she will get it back. In this way, the mechanism can be implemented in the current crowdsourcing platform such as current MTurk or CrowdFlower.

No existing crowdsourcing platform has allowed to charge their workers in advance, however, charging before providing services by a platform is widely used today.

The deposit here is like cash pledge, guarantee deposit, antecedent money or deposit in security that are used widely in today's life. For example, Pillow, a vacation rental management platform, recommends that all owner's preset a security deposit in line with the repair and/or replacement of valuable items in their property. In addition, the prepay mobile phone that charges from users before providing services is found across the world. The prepaid financial services[6] provide a prepaid travel card to offer a safe alternative to carrying money aboard. SmartGridCIS[7] optimizes prepay offering and serves as a platform to enable broader energy efficiency initiatives. These are all examples of charging users in advance, and then providing users with services by the platforms.

4 Crowdsourcing Community Improvement

In Section 3, we provide the details regarding how to find the proper reward-penalty function pairs that are related to workers' reporting tuples. In this section we perform theoretical analysis on the reward-penalty function pairs, and examine properties of the proposed mechanism that can be exploited to benefit the whole crowdsourcing community.

4.1 Personal Order Value

In the aforementioned description of the proposed mechanism, we mainly focus on a single task for each worker. While in this part, we will show that this mechanism can be used to control a worker's long term performance (e.g. approval rate).

Here we use term *personal order value* to refer to the value of order k in Eq. 11. We propose to assign each worker a personal order value k according to her approval rate. Value k can indirectly influences on the amount of her reward, penalty and expected payment since Eqs. 10 and 11 take variable k as a parameter. An example function of computing personal order value k with respect to *approval rate* is given by:

$$k = \frac{1}{approval\ rate} + 1, \tag{14}$$

where $approval\ rate \in (0, 100\%]$, and k can take continuous values. Clearly, the personal order takes the least value $k = 2$ when $approval\ rate$ is largest, and vice versa.

The computing process of worker n's $approval\ rate$ after T reportings is given in ALGORITHM 1. Line 3 takes into account the number of past reporting records in calculating the $approval\ rate$. For example, a 100% approval rating from just a single review and a 100% approval rating from 100 reviews should carry different levels of confidence. Lines 4-8 ensure that $approval\ rate$ is most determined by the last few reportings. What's more, the larger the changing rate, the greater the influence. Lines 9-11 ensure that the value of $approval\ rate$ is larger than or equal to the lower limit α. This is useful as a new worker with just several reportings will always make $approval\ rate = 0$, and the existence of lower limit helps to keep the new workers engaged in the crowdsourcing community.

6. http://www.prepaidfinancialservices.com
7. http://smartgridcis.com

Algorithm 1: Computing process of worker n's $approval\ rate$ after T reportings

Input: lower limit $\alpha \in (0, 1)$, changing rate $\eta \in (0, 1)$, benchmark solutions $\{\widehat{y}_t|_{t=1}^{T}\}$, worker n's reporting solutions $\{y_{n,t}^{*}|_{t=1}^{T}\}$
Output: $approval\ rate$

```
1  Initialize α, η, approval rate = 1;
2  for t = 1; t ≤ T; t++ do
3      η' = η + (1 − η)(1 − 1/t);
4      if y*_{n,t} ≠ ŷ_t then
5          approval rate = η' × approval rate;
6      else
7          approval rate = η' × approval rate + (1 − η');
8      if approval rate < α then
9          approval rate = α;
10 return approval rate.
```

The control of personal order value over workers' long term performance is ensured by Proposition 1. Proposition 1 suggests that for two workers with different personal value orders $k_1 > k_2$, the worker with k_2 will get a larger expected payment than the worker with k_1 for the same belief value. Since then, each worker is obliged to perform the best in the long term. Note that, a worker's personal order may be changing over time, and personal order values of workers are different. However, the incentive compatible property and other properties of the mechanism that we have discussed in the previous part still hold.

Proposition 1. *Function* $g_c(k) = \frac{-2\,ck + 2^k c^k + k - 1}{2^k - k - 1}$ *is a monotonically decreasing function of* k *in* $k \in [2, +\infty)$ *if* $c \in (0.5, 1)$.

Proof 1. For convenience we replace symbol k with x, and $2c$ with c, then an equivalent function of $g_c(k)$ is given by:

$$f_1(x) = \frac{c^x - c \cdot x + x - 1}{2^x - x - 1}, \tag{15}$$

where $1 < c < 2$, and the independent variable x is in interval $(2, +\infty)$. In the following, we prove that $f(x)$ is a decreasing function.

For simplicity, we start with looking at the asymptotic behavior. It is clear that, as x tends to be positive infinity, this function tends to $\left(\frac{c}{2}\right)^x$, which is an exponentially-decaying function for $1 < c < 2$.

If we take the derivative of this function, we have

$$\frac{(2c)^x \ln c - (2c)^x \ln 2}{2^{2x}} = \left(\ln \frac{c}{2}\right) \frac{(2c)^x}{2^{2x}}, \tag{16}$$

for which the logarithmic factor is negative, making the derivative function negative.

By taking the derivative of the complete expression of the function, we have

$$f_1'(x) = \frac{\ln \frac{c}{2}\,(2c)^x - ((1+x)(\ln c) - 1)c^x - ((c - 1 + \ln 2) + (c-1)(\ln 2)\,x)2^x + (c-2)}{(2^x - x - 1)^2}. \tag{17}$$

The denominator is always positive for $x > 2$, so we only focus on the numerator. The first term is negative and the last term is also negative for $1 < c < 2$. The other assembled terms are linear functions times exponential growth functions. For the third term, we find that

$$(c-1+\ln 2)+(c-1)(\ln 2)\,x > 0 \Rightarrow x > -\frac{c-1+\ln 2}{(c-1)\ln 2}, \quad (18)$$

so the third term in the numerator of the derivative is negative for the values of x and c under discussion.

In the second term, it is relatively complex for the factor $(1+x)(\ln c)-1$: it is positive for $x > \frac{1}{\ln c} - 1 = \tilde{x}$, so the second term in the numerator of the derivative is positive for $0 < x < \tilde{x}$. This is not an issue for $e^{1/3} < c < 2$ (for which $\tilde{x} < 2$), but otherwise we must examine the behavior of the term for smaller values of c. What we find is that the second term is always less than $+1$ for $0 < x < \tilde{x}$, so it is always less positive than the third term in the numerator is negative for $x > 2$.

Consequently, we can conclude that the entire numerator is negative for $x > 2$ and $1 < c < 2$. In other words, $f_1'(x) < 0$ for $x > 2$ and $1 < c < 2$. Hence, function $g_c(k)$ is a monotonically decreasing function of k in $k \in [2, +\infty)$ if $c \in (0.5, 1)$. □

A worker's approval rate is derived from her historical data, which is relative to her experiences and professional knowledge, etc. It means to some extent the different probabilities of workers to give the right solution. According to Eq. 14, a worker's approval rate determines her personal order value. From Eq. 10 and 11, a worker's personal order value and her reporting belief determine her reward, penalty and expected payment. Eq. 12 indicates that a worker's expected payment influences the final solution. Therefore, we can draw the conclusion that workers' long term performance and their reportings together determines the final solution.

Apart from ensuring workers' long term performance, another three benefits that personal order value brings about are as follows: 1) Saving cost of requesters. A worker with a low approval rate is paid less for her reporting, which can reduce the cost of the requester. Consequently, compared with a single reward function or a single reward-penalty function pair, the proposed mechanism is more effective to control the cost of the requesters, and keep the property of incentive compatible principle meanwhile; 2) Improving reporting quality. To increase the approval rate, a worker will attempt to ensure the quality of her each reporting; and 3) A requester can use the cost she saved to publish more tasks or collect more reportings on the crowdsourcing platform.

4.2 Improvement of Crowdsourcing Platform

Compared to other three key evaluation indices of crowdsourcing community, platform improvement is often overlooked to some extent [7]. However, it may exert tremendous influence over the workers, requesters and crowdsourcing platforms. Still taking MTurk as an example[8], in 2012, MTurk decided not to accept new international worker accounts any more, as the quality of work declined steadily and the requesters begun to complain about or leave MTurk. This change lies in the fact that badly-behaved international workers flooded into MTurk just for money, and MTurk cannot effectively control the reporting quality of workers from various countries. However, it sacrificed long-term development of the crowdsourcing community. The reason lies in the fact that professional workers overseas cannot make money on MTurk any more, and the requester cannot get enough professional workers to undertake the published tasks within the time constraint.

8. http://turkrequesters.blogspot.com/2013/01/the-reasons-why-amazon-mechanical-turk.html

Proposition 2. *With fixed cost of the requester, the professional workers under the proposed mechanism can earn more, while badly-behaved workers earn less for each task, compared with current crowdsourcing platforms such as MTurk and CrowdFlower.*

Proof 2. The expected gain function $g_k''(c)$ for all $k \geq 2$ is a convex function of variable c, which is guaranteed by the following second derivative:

$$g_k''(c) = \frac{2^k k(k-1)c^{k-2}}{2^k - k - 1} > 0. \quad (19)$$

In addition, $g_k(0.5) = 0$ for all $k \geq 2$.

While in current crowdsourcing platforms such as MTurk and CrowdFlower, once a reporting type is judged as right, the requester should pay the same money to the workers. In this settings, for a task with belief value c, the expected gain function can be written as

$$f(c) = c \cdot \eta - (1-c) \cdot 0 = \eta c, \quad (20)$$

where η is just a constant coefficient. In addition, we have $f(0.5) > 0$ and $f''(c) = 0$.

Suppose the total amount of money that the requester payed for the workers are fixed, and variable c is a uniform random number in range $(0.5, 1.0)$, then we can obtain the following equation:

$$\int_{0.5}^{1} g_k(c)\mathrm{d}c = \int_{0.5}^{1} f(c)\mathrm{d}c. \quad (21)$$

Then, for simplicity, in the following we use symbol $g(x)$ to denote $g_k(c)$, and use symbol $f(x)$ to denote $f(c)$. The constraints can be summarized as follows: Given two functions $g(x) > 0$ and $f(x) > 0$ on $x \in [a, b]$:

$$\begin{cases} g''(x) > 0 \\ f''(x) = 0 \\ g(a) < f(a) \\ \int_a^b g(x)\mathrm{d}x = \int_a^b f(x)\mathrm{d}x \end{cases}. \quad (22)$$

As $\int_a^b f(x)\mathrm{d}x = \int_a^b g(x)\mathrm{d}x$, we have $\int_a^b f(x) - g(x)\mathrm{d}x = 0$. $f(x)$ and $g(x)$ are differentiable, and hence they must be continuous. If $f(x)$ and $g(x)$ do not intersect, $f(x) - g(x)$ does not change sign. Then, $f(x) - g(x) > 0$ or $f(x) - g(x) < 0$ for all $x \in (a, b)$ and the integration will not be 0 which is a contradiction. Therefore, $f(x)$ and $g(x)$ surely intersect in range (a, b).

Suppose $f(x)$ and $g(x)$ have more than two intersections in range (a, b), of which the nearest two intersections to $x = a$ are at $x = s_1$ and $x = s_2$. This means we have $(f -$

$g)(s_1) = 0$ and $(f-g)(s_2) = 0$, so there must exist $x = s_3$, such that $(f-g)'(s_3) = 0$. Note that $(f-g)''(x) < 0$, and hence $(f-g)'(x)$ decreases in range (a, b). Then, we have $(f-g)'(x) > 0$ in range (a, s_3). As $(f-g)(s_1) = 0$, it holds that $(f-g)(a) < 0$, which is a contradiction. Therefore, $g(x)$ and $f(x)$ must have and only have one intersection on interval (a, b). Then suppose the only one intersection $g(x)$ and $f(x)$ is at $x = s$. As $g(a) < f(a)$, we have $g(x) < f(x)$ in range (a, s). As $\int_a^b f(x) - g(x)\mathrm{d}x = 0$, we have $g(x) > f(x)$ in range (s, b).

In summary, the following conclusions hold: 1) $g(x)$ and $f(x)$ must have and only have one intersection on interval (a, b); and 2) If the x-coordinate of the intersection is $x = s$, we have:

$$\begin{cases} g(x) < f(x), & \text{if} \quad x \in (a, s) \\ g(x) > f(x), & \text{otherwise} \end{cases}. \tag{23}$$

For professional workers, the belief values for most tasks are larger than badly-behaved workers. As a result, under the proposed mechanism, the professional workers can earn more, while badly-behaved workers earn less for an average task than in current crowdsourcing platforms such as MTurk and CrowdFlower. □

By contrast, the proposed mechanism can attract professional workers into crowdsourcing community and squeeze out badly-behaved ones, which benefits the platform improvement. This property is ensured by Proposition 2. What we need do is to make use of the personal order value k, making the payments to badly-behaved workers lower than the thresholds they can accept, and the payments to professional workers higher than the thresholds. MTurk gives the requesters the power to manually block some badly-behaved workers that they have observed. However, MTurk has no inherent mechanism to squeeze out badly-behaved workers automatically. What's more, blocking badly-behaved workers by requesters is highly subjective and biased, which the crowdsourcing community should strive to avoid.

4.3 Latency Control

In [17], [6] and current crowdsourcing platforms, if a worker's evaluation reliability[9] is lower than a certain *threshold value*, she would skip this difficult task rather than giving an unbelievable solution at a guess. In the case of skipping, even though the worker has spent time and efforts on it, she gains no payment; meanwhile the requester cannot successfully collect workers' reportings in time for the difficult task. But a task skipped by most workers will lag behind, which will lead to excessive latency. So excessive latency is a lose-lose situation, where interests of both participants are violated.

While under the proposed mechanism, when a worker's evaluation reliability for a difficult task is very low, she can report a very small belief value (close to 0.5) along with her reporting type for a tiny payment. Meanwhile, the requester can collect enough reportings for each task. Then, the difficult tasks will not lag behind and excessive latency would not occur. That is to say, the proposed mechanism can benefit both of the workers and requesters by latency control.

It does not mean that if the workers do not know the answer to a task, they can always get a tiny reward by give a random solution and a belief value near 0.5. The premise of reward, no matter the reward is tiny or large, is a reporting solution that is judged right. If the worker totally has no idea of the solution to a task, she would give the right or wrong solution at equal probability. In the proposed mechanism, the penalty is larger than the reward if the reporting belief is the same. So the worker would not only get reward in general, but also pay a much larger penalty. So the best action for the worker is to skip this task or give a solution with a belief value 0.5. Note that both of these two actions lead to the same results, i.e., gain nothing, as the worker provides no useful information to the requester to obtain a right final solution. It should be noted that the case where the worker has no idea of a solution to a difficult task is very rare. It often happens that the worker can give a self-convinced right solution, however, the worker is not quite sure of the solution. When faced with a difficult task, the proposed mechanism suggests that the workers can report a very small belief value (close to 0.5) along with her reporting type for a tiny payment. As the real belief value of a difficult task is close to 0.5, the worker's estimated belief value is close to 0.5 in general. According to our proposed mechanism, the worker should report the estimated belief value truthfully. Of course, the worker would just get a tiny payment. Note that this is quite different from the former case where a worker that totally has no idea of the solution gives a random solution. In the latter case, the worker still provides useful information, although the usefulness is very small. In fact, this is a strategy to reduce the risk of judged wrong when a worker is not sure of the solution for a difficult task.

5 Experiments

In addition to performing theoretical analysis in the last section, we also conduct simulation experiments to further validate the proposed mechanism. We cannot implement and run the proposed mechanism on existing crowdsourcing platforms (e.g. MTurk or CrowdFlower) as none of existing crowdsourcing platforms has incorporated penalty into their current mechanisms. Most advantages of the proposed mechanism are proved in theory. Although our experiments exist some limitations, it can still validate the advantages to some extent.

5.1 Experiment Settings

Inspired by [8], [5], [3], in generating the desired synthetic data, we only consider the simplest case, where tasks' real types and beliefs are evenly distributed, and workers' biases follow truncated normal distribution.

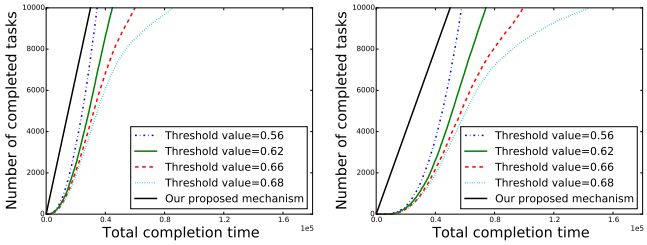
The details to generate the synthetic data are as follows. A binary-type task's real type takes $y = +1$ with probability of 0.5, and $y = -1$ with the same probability. The probability that a worker gives the right type for this task follows a

9. Evaluation reliability is represented by belief value in this paper, and it is related to the inherent difficulty of the task, the worker's different experiences and professional knowledge, etc.

uniform distribution in range $[0.5, 1]$. The truncated normal distribution can be considered as the standard normal distribution within $[-0.2, 0.2]$. The specific process of generating a reporting tuple of a worker (with bias value b) on a task (with real type $y = +1$ and real belief c_0) are given by

$$\langle \text{type}, belief \rangle = \begin{cases} \langle -1, 1 - (c_0 + b) \rangle, & \text{if } c_0 + b < 0.5 \\ \langle +1, 1 \rangle, & \text{if } c_0 + b > 1 \\ \langle +1, c_0 + b \rangle, & \text{else} \end{cases} . \quad (24)$$

In this simulation, we generate the synthetic data of 100 workers on $10,000$ tasks. In addition, for simplicity, we assume workers cost the same time to complete a task and report the solution.



(a) 3 workers to each task (b) 5 workers to each task

Fig. 4: Comparison in latency control between our proposed mechanism and current crowdsourcing with different threshold values.

5.2 Latency Control

Fig. 4 shows the number of completed crowdsourcing tasks with respect to total completion time. It can be seen that the proposed mechanism outperforms the mechanism in [17], [6] and current crowdsourcing platform, where a worker only gives her reporting when her belief value for the coming task is larger than the threshold value. For simplicity, if the number of workers that have given reportings on a task has reached a limit (e.g. 3 or 5), we will consider this task is completed. For example, in Fig. 4a, the total completion time of the current mechanism for $10,000$ tasks grows faster and faster as the threshold value increases from 0.56 to 0.68 (i.e., $0.56, 0.62, 0.66$ and 0.68), and it is larger than the total completion time of the proposed mechanism all the time. This phenomenon in Fig. 4b is more evident than that in Fig. 4a, as the number of workers to each tasks changes from 3 to 5. Therefore the proposed mechanism can avoid the occurrence of excessive latency effectively.

5.3 Cost and Platform Improvement

Fig. 5 illustrates the expected payment that a worker can get from the requester with different reporting belief values. Clearly, for each personal order value, there exists a threshold of reporting belief value. If a worker's belief value for a task is lower than this value, she will earn less under the proposed mechanism than in current crowdsourcing, vice versa. What's more, the larger the personal order value ($2, 5, 9$ and 12), the more obvious the trend. This results validate that the proposed mechanism can 1) reduce the cost of the requester because of low expected payment for low quality reportings; and 2) attract professional workers into crowdsourcing community and squeeze out badly-behaved ones, which benefits the platform improvement. Based on the above results, we can also draw the conclusion that the proposed mechanism can complete more tasks with the same cost and total completion time.

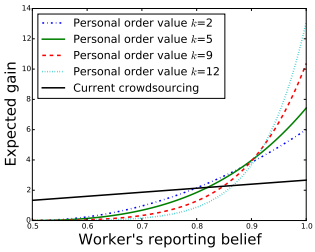


Fig. 5: Expected gain pay- ment with different report- ing belief values.

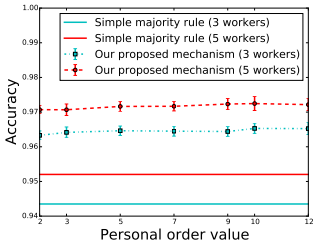


Fig. 6: Accuracy with different personal order values.

5.4 Quality

Fig. 6 shows the accuracy comparison with different personal order values. We can see that 1) for both of simple majority rule and our proposed mechanism, a larger number of workers to each task will lead to higher accuracy; 2) with a fixed number of workers to each task, our proposed mechanism performs better than the simple majority rule; and 3) for our proposed mechanism, as the personal order value increases, the accuracy has a slight ascending tendency. Therefore, the experimental results demonstrates that our proposed mechanism can help to improve the quality of reportings in crowdsourcing community.

The boundary value of the real belief value at $c = 0.5$ is quite different. If the real belief value of a task is exactly 0.5, according to the definition of the real belief value, the task has no a real solution, or this task cannot be cope with crowdsourcing. In fact, the discussed bound value of 0.5 cannot be reached in practical cases in crowdsourcing. If we set the belief value x=0.5, the simulation results obtained from several simulations for the same task will be different to a very large extent.

6 Related Works

Among the four key evaluation indices of crowdsourcing, quality can be considered as the core. Essentially, it is served by the remaining three (i.e. cost, latency and platform improvement.). The quality of reportings is important because a task's final truthful solution is generated based on the collected reportings from workers. As a result, we start with quality as the breakthrough point. An important issue in crowdsourcing community is how to estimate the worker proficiency (i.e. the probability that she correctly evaluates the tasks in general) [14] or her evaluation reliability for a specific task (i.e. the probability that a worker's solution for a specific task is correct) [15], [16]. One natural approach is to use *gold standard* method [29], [7], which however, works not well in heterogenous crowdsourcing [28]. Then, studies in [30], [14] assume that the tasks can be categorized into several topics and workers differ in their abilities with different topic tasks. Based on this assumption, what we need do is just to estimate a worker's ability on a specific

topic. But it relies heavily on topic categorization results. [31], [32] developed statistical post-process techniques, which however, are proper only with repeated reportings from each worker in a short period. But it seems impractical. In fact, the current crowdsourcing platforms such as MTurk and CrowdFlower adopt a worker's history approval rate to decide whether to assign her a task or not. But it cannot forbid malicious workers to perform as normal to gain high approval rate firstly, and afterwards cheat in order to gain more payments. However, direct monitoring of workers' effort and accuracy in performing tasks is difficult. An alternative is to induce workers to make good evaluations and report them truthfully, thereby achieving high quality.

Instead of indirectly estimating the worker proficiency or her reporting's reliability, we assume that a worker knows her reporting's reliability for a specific task, and try to design a mechanism to stimulate every worker to report truthfully. Similar works are theoretical development of mechanisms (such as Peer Prediction [15] and Bayesian Truth Serum [16]), and the deployment of simple and intuitive mechanisms (such as the Output Agreement mechanism [19]). Different from these works, the proposed mechanism may reward or punish a worker, instead of giving only nonnegative payment in [15], [16], [19]. It shows in [17] that negative payments (i.e. penalty in this paper) can be used to make workers with quality lower than the quality threshold choose to not to participate, while those above continue to participate and invest effort. But it will lead to excessive latency if difficult tasks are skipped by most workers lag behind. [33] tries to encourage workers to devote effort to make good evaluations, as well as to truthfully report their evaluations. However, it still requires prior knowledge of every task, which is hard to get in practice. Therefore, it still cannot identify malicious workers effectively. Compared with [14], [30], the proposed mechanism pays less attention on how to learn and predict worker's performance information, and workers are no longer required to reveal their effort information to the platform. In addition, the proposed mechanism takes into account the incentive compatible principle [16], [9], gradients [20] and bounds [26] of reward-penalty function pairs. Further more, we find a series of proper reward-penalty function pairs. As a result, the proposed mechanism finally aligns with the incentives of three stakeholders in crowdsourcing. These differences endow the proposed mechanism with some desired properties to benefit the long-term development of the whole crowdsourcing community.

7 Conclusion

In this paper, we have demonstrated that a crowdsourcing community involves the interests of the three stakeholders, namely requester, worker and crowdsourcing platform, and the incentives among them always conflict with each other. We have proposed and verified the hypothesis that all workers believe that in most cases they observe the real solution of each task perturbed only by unbiased noise, and design a crowdsourcing mechanism, encompassing a series of proper reward-penalty function pairs and workers' personal order values, to align the interests of different stakeholders, which has been validated by the theoretical analysis and experimental results. This work can help to relieve the platform and requesters of crowdsourcing community from monitoring workers' efforts and capacities in performing crowdsourcing tasks, save the costs of requesters, and attract more professional workers to the crowdsourcing platforms. It can accelerate the long-term development of the whole crowdsourcing community

For the future work, we will study the following potential directions: 1) We will build up a small crowdsourcing platform based on the proposed mechanism to test and promote the proposed mechanism; 2) We will further adapt the proposed mechanism to make it work properly within limited total budget; 3) We will extend the proposed mechanism to directly deal with multiple type tasks; and 4) We will also study security and privacy aspects of crowdsourcing to facilitate wide-deployment of crowdsourcing [34].

Acknowledgment

This work was supported in part by the National Science Foundation of China (No.61472047 and 61571066) and National Key R&D Program of China (Grant No.2018YFB1004800). Shangguang Wang is the corresponding author.

References

[1] B. Guo, C. Chen, D. Zhang, Z. Yu, and A. Chin, "Mobile crowd sensing and computing: when participatory sensing meets participatory social media," *IEEE Communications Magazine*, vol. 54, no. 2, pp. 131–137, 2016.

[2] R. Jurca and B. Faltings, "Error rate analysis of labeling by crowdsourcing," in *Proceedings of the 30th International Conference on Machine Learning Workshop (ICML)*, pp. 1–19, MIT Press, 2013.

[3] Y. Gao, Y. Chen, and K. J. R. Liu, "On cost-effective incentive mechanisms in microtask crowdsourcing," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 7, pp. 3–15, 3 2015.

[4] J. Ren, Y. Zhang, K. Zhang, and X. Shen, "Exploiting mobile crowdsourcing for pervasive cloud services: challenges and solutions," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 98–105, 2015.

[5] B. Aydin, Y. Yilmaz, Y. Li, Q. Li, J. Gao, and M. Demirbas, "Crowdsourcing for multiple-choice question answering," in *Proceedings of the 26th Innovative Applications of Artificial Intelligence (IAAI)*, pp. 2946–2953, AAAI, 2014.

[6] N. B. Shah and D. Zhou, "Double or nothing: Multiplicative incentive mechanisms for crowdsourcing," in *Proceedings of the 28th Advances in Neural Information Processing Systems (NIPS)*, pp. 1–9, MIT Press, 2015.

[7] G. Li, J. Wang, Y. Zheng, and M. Franklin, "Crowdsourced data management: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. PP, no. 99, pp. 1–23, 2016.

[8] P. Chandra, Y. Narahari, and D. Mandal, "Novel mechanisms for online crowdsourcing with unreliable, strategic agents," in *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pp. 1256–1262, AAAI, 2015.

[9] N. Chen, X. Deng, B. Tang, and H. Zhang, "Incentives for strategic behavior in fisher market games," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pp. 453–459, AAAI, 2016.

[10] M. Al-Ayyoub and H. Gupta, "Truthful spectrum auctions with approximate social-welfare or revenue," *IEEE/ACM Transactions on Networking (TON)*, vol. 22, no. 6, pp. 1873–1885, 2014.

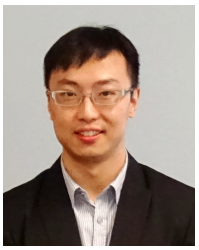
[11] M. Li, P. Li, M. Pan, and J. Sun, "Economic-robust transmission opportunity auction in multi-hop wireless networks," in *Proceedings of 32nd IEEE International Conference on Computer Communications (INFOCOM)*, pp. 1842–1850, IEEE, 2013.

[12] J. Wang, S. Faridani, and P. Ipeirotis G., "Estimating the completion time of crowdsourced tasks using survival analysis models," in *Proceedings of the 4th Crowdsourcing for Search and Data Mining (CSDM)*, pp. 1–4, ACM, 2011.

[13] J. Vuurens, A. P. de Vries, and C. Eickhoff, "How much spam can you take? an analysis of crowdsourcing results to increase accuracy," in *Proceedings of 34th ACM SIGIR Workshop on Crowdsourcing for Information Retrieval*, pp. 21–26, ACM, 2011.
[14] E. Simpson and S. Roberts, "Bayesian methods for intelligent task assignment in crowdsourcing systems," in *Decision Making: Uncertainty, Imperfection, Deliberation and Scalability*, pp. 1–32, Springer, 2015.
[15] N. Miller, P. Resnick, and R. Zeckhauser, "Eliciting informative feedback: The peer-prediction method," *Management Science*, vol. 51, no. 9, pp. 1359–1373, 2005.
[16] G. Radanovic and B. Faltings, "A robust bayesian truth serum for non-binary signals," in *Proceedings of the 27th AAAI Conference on Artificial Intelligence*, pp. 833–839, AAAI, 2013.
[17] J. Witkowski, Y. Bachrach, P. Key, and D. C. Parkes, "Dwelling on the negative: Incentivizing effort in peer prediction," in *Proceedings of the 1st AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*, pp. 190–197, AAAI, 2013.
[18] B. W. Silverman, "Density estimation for statistics and data analysis," *Monographs on Statistics and Applied Probability*, vol. 26, 1986.
[19] L. Von Ahn and L. Dabbish, "Designing games with a purpose," *Communications of the ACM*, vol. 51, no. 8, pp. 58–67, 2008.
[20] C. Vullioud, F. Clément, T. Scott-Phillips, and H. Mercier, "Confidence as an expression of commitment: Why misplaced expressions of confidence backfire," *Evolution and Human Behavior*, vol. 38, no. 1, pp. 1–37, 2016.
[21] G. Radanovic and B. Faltings, "Incentives for subjective evaluations with private beliefs," in *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pp. 1014–1020, AAAI, 2015.
[22] J. Witkowski and D. C. Parkes, "Peer prediction without a common prior," in *Proceedings of the 13th ACM Conference on Electronic Commerce (EC)*, pp. 964–981, ACM, 2012.
[23] J. Witkowski and S. Seuken, "Incentive-compatible escrow mechanisms," in *Proceedings of the 25th AAAI Conference on Artificial Intelligence*, pp. 751–757, AAAI, 2011.
[24] D. Berend and A. Kontorovich, "Consistency of weighted majority votes," in *Proceedings of the 24th Advances in Neural Information Processing Systems (NIPS)*, pp. 3446–3454, MIT Press, 2014.
[25] X. Sun, H. Shen, X. Cheng, and Y. Zhang, "Market confidence predicts stock price: Beyond supply and demand," *PLOS ONE*, vol. 11, no. 7, p. e0158742, 2016.
[26] S. Liang and J. Zhang, "Positive solutions for boundary value problems of nonlinear fractional differential equation," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 71, no. 11, pp. 5545–5550, 2009.
[27] D. Prelec and S. Seung, "An algorithm that finds truth even if most people are wrong," *Unpublished manuscript*, 2006.
[28] H. Zhang and M. Sugiyama, "Task selection for bandit-based task assignment in heterogeneous crowdsourcing," in *Proceedings of the 20th Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, pp. 164–171, AAAI, 2015.
[29] X. Liu, M. Lu, B. C. Ooi, Y. Shen, S. Wu, and M. Zhang, "Cdas: A crowdsourcing data analytics system," *Proceedings of the VLDB Endowment*, vol. 5, no. 10, pp. 1040–1051, 2012.
[30] P. G. Ipeirotis, F. Provost, V. S. Sheng, and J. Wang, "Repeated labeling using multiple noisy labelers," *Data Mining and Knowledge Discovery*, vol. 28, no. 2, pp. 402–441, 2014.
[31] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy, "Learning from crowds," *Journal of Machine Learning Research*, vol. 11, no. 4, pp. 1297–1322, 2010.
[32] D. Zhou, Q. Liu, J. C. Platt, and C. Meek, "Aggregating ordinal labels from crowds by minimax conditional entropy," in *Proceedings of the 31st International Conference of Machine Learning (ICML)*, pp. 262–270, MIT Press, 2014.
[33] A. Dasgupta and A. Ghosh, "Crowdsourced judgement elicitation with endogenous proficiency," in *Proceedings of the 22nd International Conference on World Wide Web (WWW)*, pp. 319–330, ACM, 2013.
[34] K. Yang, K. Zhang, J. Ren, and X. Shen, "Security and privacy in mobile crowdsourcing networks: challenges and opportunities," *IEEE Communications Magazine*, vol. 53, no. 8, pp. 75–81, 2015.

Jinliang Xu received the bachelor's degree in electronic information science and technology from Beijing University of Posts and Telecommunications in 2014. Currently, he is a Ph.D. candidate in computer science at the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. His research interests include Mobile Cloud Computing, Service Computing, Information Retrieval, and Crowdsourcing.

Shangguang Wang received his PhD degree at Beijing University of Posts and Telecommunications in 2011. He is an associate professor at the State Key Laboratory of Networking and Switching Technology (BUPT). He has published more than 100 papers, and played a key role at many international conferences, such as general chair and PC chair. His research interests include service computing, cloud computing, and mobile edge computing. He is a senior member of the IEEE, and the Editor-in-Chief of the International Journal of Web Science.



Ning Zhang received the Ph.D degree from University of Waterloo in 2015. Since May 2015, he has been a postdoc research fellow at BBCR lab in University of Waterloo. He is now an associate editor of International Journal of Vehicle Information and Communication Systems and a lead guest editor of International Journal of Distributed Sensor Networks. He is the recipient of the Best Paper Award at IEEE Globecom 2014 and IEEE WCSP 2015. His current research interests include next generation wireless networks, software defined networking, vehicular networks, and physical layer security.

Fangchun Yang received his Ph.D. degree in communication and electronic system from the Beijing University of Posts and Telecommunication in 1990. He is currently a professor at the Beijing University of Posts and Telecommunication, China. His research interests include network intelligence and communications software. He is a fellow of the IET.

Xuemin (Sherman) Shen (M97, SM02, F09) received the B.Sc. (1982) degree from Dalian Maritime University (China) and the M.Sc. (1987) and Ph.D. degrees (1990) from Rutgers University, New Jersey (USA), all in electrical engineering. He is a Professor and University Research Chair, Department of Electrical and Computer Engineering, University of Waterloo, Canada. He is also the Associate Chair for Graduate Studies. Dr. Shen's research focuses on resource management in interconnected wireless/wired networks, wireless network security, social networks, smart grid, and vehicular ad hoc and sensor networks. He is an elected member of IEEE ComSoc Board of Governor, and the Chair of Distinguished Lecturers Selection Committee. Dr. Shen served as the Technical Program Committee Chair/Co-Chair for IEEE Globecom'16, IEEE Infocom'14, IEEE VTC'10 Fall, the General Chair for ACM Mobihoc'15, the Symposia Chair for IEEE ICC'10, the Tutorial Chair for IEEE VTC'11 Spring and IEEE ICC'08, the Technical Program Committee Chair for IEEE Globecom'07, the General Co-Chair for Chinacom'07 and QShine'06, the Chair for IEEE Communications Society Technical Committee on Wireless Communications, and P2P Communications and Networking. He also serves/served as the Editor-in-Chief for IEEE Network, IEEE Internet of Things Journal, Peer-to-Peer Networking and Application, and IET Communications; a Founding Area Editor for IEEE Transactions on Wireless Communications; an Associate Editor for IEEE Transactions on Vehicular Technology, Computer Networks, and ACM/Wireless Networks, etc.; and the Guest Editor for IEEE JSAC, IEEE Wireless Communications, IEEE Communications Magazine, and ACM Mobile Networks and Applications, etc. Dr. Shen received the Excellent Graduate Supervision Award in 2006, and the Outstanding Performance Award in 2004, 2007 and 2010 from the University of Waterloo, the Premiers Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada, and the Distinguished Performance Award in 2002 and 2007 from the Faculty of Engineering, University of Waterloo. Dr. Shen is a registered Professional Engineer of Ontario, Canada, an IEEE Fellow, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, and a Distinguished Lecturer of IEEE Vehicular Technology Society and Communications Society.