Incentives for Effort in Crowdsourcing Using the Peer Truth Serum

GORAN RADANOVIC and BOI FALTINGS, Artificial Intelligence Laboratory, EPFL
RADU JURCA, Google

Crowdsourcing is widely proposed as a method to solve a large variety of judgment tasks, such as classifying website content, peer grading in online courses, or collecting real-world data. As the data reported by workers cannot be verified, there is a tendency to report random data without actually solving the task. This can be countered by making the reward for an answer depend on its consistency with answers given by other workers, an approach called *peer consistency*. However, it is obvious that the best strategy in such schemes is for all workers to report the same answer without solving the task.

Dasgupta and Ghosh [2013] show that, in some cases, exerting high effort can be encouraged in the highest-paying equilibrium. In this article, we present a general mechanism that implements this idea and is applicable to most crowdsourcing settings. Furthermore, we experimentally test the novel mechanism, and validate its theoretical properties.

Categories and Subject Descriptors: J.4 [**Social and Behavioral Sciences**]: Economics

General Terms: Human Factors, Economics, Measurement

Additional Key Words and Phrases: Crowdsourcing, mechanism design, peer prediction

ACM Reference Format:
Goran Radanovic, Boi Faltings, and Radu Jurca. 2016. Incentives for effort in crowdsourcing using the peer truth serum. ACM Trans. Intell. Syst. Technol. 7, 4, Article 48 (March 2016), 28 pages.
DOI: http://dx.doi.org/10.1145/2856102

1. INTRODUCTION

Crowdsourcing is an effective method for eliciting information for a variety of tasks, such as classifying website content, peer grading in online courses, or gathering data about the real world. As workers are recruited anonymously through the Internet, a major issue is how to ensure that their answers are accurate. There are two aspects to this problem: selecting workers with the best abilities for the task, and getting them to invest their best effort to obtain the most accurate answers. A large body of work has been devoted to the issue of how to select workers for tasks, either through learning their quality and assigning them to the most appropriate tasks, or through incentive schemes (mechanisms) that encourage self-selection of the most competent workers [Singla and Krause 2013; Witkowski et al. 2013].

The other issue is to get workers to invest sufficient effort. An obvious strategy that maximizes workers' profit is to just provide arbitrary answers without even solving the tasks. This is clearly observable in practice, in which crowdsourcing tasks attract a significant portion of workers that provide random answers.

This work is supported by Nano-Tera.ch as part of the Opensense2 project.
Authors' addresses: G. Radanovic and B. Faltings, Artificial Intelligence Laboratory, EPFL, CH-1015 Lausanne, Switzerland; emails: {goran.radanovic, boi.faltings}@epfl.ch; R. Jurca, Google, CH-8002 Zurich, Switzerland; email: radu.jurca@gmail.com.

DOI: http://dx.doi.org/10.1145/2856102

We propose to overcome this problem by making payments depend on the accuracy of answers, so that only workers that provide actual work are rewarded. This will complement existing methods for filtering answers and workers such as gold tasks, but will also help because workers that do not provide useful answers are discouraged from participating in the tasks in the first place. At first glance, this appears to be impossible, as the system itself cannot verify the accuracy of answers given by workers.

Remarkably, this problem can be solved elegantly by setting up a *game* among workers, called a *mechanism* in game theory. Instead of paying fixed payments per answer or hour of time, the rewards are calculated as a function of a worker's answer and the answers given by other workers. The game is designed so that the strategies that carry the highest expected reward require workers to solve the tasks, whereas random answers will, on average, produce no reward.

This approach has been tried with success on platforms such as Amazon Mechanical Turk (AMT). Harris [2011] considered the task of screening résumés for a job description. A scheme in which payments depend on the agreement of answers with those of a human resources expert provided significant improvements in accuracy. Shaw et al. [2011] tested a large variety of reward schemes using a task of classifying the type of content present on a website, and found that schemes based on consistency of answers had the best performance. Giving rewards for agreeing with another worker has also been used in the very successful ESP game [Ahn and Dabbish 2004], in which players are rewarded for assigning the same label as a peer to an image. Kamar and Horvitz [2012] proposed to reward workers based on comparison of the answers with the aggregate obtained from the crowd. Huang and Fu [2013b] investigated the peer-consistency incentive scheme using a task of counting nouns in a list of 30 English words. Workers were rewarded with a bonus whenever their answer agreed with that of a single, randomly chosen peer. They found that this increases accuracy more than comparing against a gold standard. The same authors also showed that social pressure can further increase accuracy [Huang and Fu 2013a]. Faltings et al. [2014b] showed a modified version of peer consistency, called the *peer truth serum*, that allows the answer distribution to be biased, and showed that it can correct anchoring bias in a counting task on AMT.

Mechanisms based on agreement of answers, also known as peer consistency, peer prediction, or output agreement, have been game-theoretically analyzed for the problem of incentivizing workers to give truthful information [Miller et al. 2005; Prelec 2004; Prelec and Seung 2006; Witkowski and Parkes 2012a, 2012b; Goel et al. 2009; Lambert and Shoham 2008; Zhang and Chen 2014; Radanovic and Faltings 2014]. The main difference is that, in crowdsourcing, obtaining an accurate answer requires costly effort, so that the difference in rewards between an accurate and inaccurate answer has to exceed the cost of this effort. With this additional condition, reward schemes that have been developed to incentivize truthful information reporting can be adapted for use in crowdsourcing.

However, game theoretic analysis shows that truthful reward schemes based on comparing individual answers necessarily require unrealistic assumptions of highly homogeneous user populations [Jurca and Faltings 2009; Radanovic and Faltings 2013; Waggoner and Chen 2013, 2014]. Even for homogeneous populations, minimal mechanisms that elicit individual answers cannot be constructed in an arbitrary context [Radanovic and Faltings 2013, 2015b]. Nonminimal mechanisms, such as the Bayesian truth serums [Prelec 2004; Witkowski and Parkes 2012b; Radanovic and Faltings 2013, 2014], are applicable in this case, but the burden of eliciting additional information restrict their practicality. Furthermore, strategies in which all workers report identical answers are always more profitable, and such behavior has been observed in user experiments ([Gao et al. 2013]). We do note that some schemes are designed for heterogeneous populations (e.g., Witkowski and Parkes [2012a]), but they require that the elicitation process has a clear temporal structure, which is often inconvenient for crowdsourcing.

These negative results can be overcome by reward schemes that depend on workers solving many tasks. Recently, Dasgupta and Ghosh [2013] have proposed a reward scheme that uses multiple tasks to achieve incentive compatibility for heterogeneous populations. The most profitable strategy for workers is to solve the tasks with their best effort, and random answers carry, on average, no reward. However, their scheme is only applicable to tasks with two possible answers, so that there cannot be any correlation between possible answer values.

Contributions

In this article, we present a novel mechanism that extends the mechanism of Dasgupta and Ghosh [2013] to allow any number of possible answers while also ensuring that a strategy profile in which all workers exert high effort and report their results truthfully is the most profitable equilibrium. Just like the mechanism in Dasgupta and Ghosh [2013], it is easy to understand but can be more broadly applied.

The mechanism combines the ideas from Dasgupta and Ghosh [2013] and Witkowski and Parkes [2013] with the Peer Truth Serum (PTS) introduced in Jurca and Faltings [2011] and Faltings et al. [2014a, 2014b]; we call it the Peer Truth Serum for Crowdsourcing (PTSC). The idea behind the mechanism is to use the distribution of reported answers from similar tasks as the prior probability of possible answers, and scale the reward given for agreement between workers with this distribution. This solves the major issue with the PTS mechanism presented in Faltings et al. [2014b], which is that the prior distribution had to be known. While a similar approach has been studied in Witkowski and Parkes [2013], its analysis did not include all of the incentive properties discussed in Dasgupta and Ghosh [2013].

Unlike the mechanism from Dasgupta and Ghosh [2013], PTSC uses its statistic (prior obtained from reports) in a nonlinear manner. When the statistic is calculated from only two *a priori* similar tasks, PTSC is equivalent to the mechanism from Dasgupta and Ghosh [2013], which requires possible answer values to be uncorrelated. However, when the sample is large, PTSC allows more significant correlations among different values. We also note that, unlike the mechanism in Dasgupta and Ghosh [2013], PTSC does not assume that any particular worker solves more than one task. Moreover, we show how to elicit a value of the mechanism's scaling parameter for which high effort and honest reporting is the most profitable equilibrium. Finally, we apply the PTSC mechanism in community (participatory) sensing and peer grading settings, and report on empirical results that validate its theoretical properties. The proofs to our formal claims can be found in the corresponding Online Appendix.

2. THE PEER TRUTH SERUM FOR CROWDSOURCING

In our crowdsourcing model shown in Figure 1, a group of workers w, p, and q solve their tasks t_w, t_p and t_q, respectively. After worker w evaluates the correct answer to the task t_w, worker w updates belief $Pr(x_p, x_q|x_w)$ regarding the evaluations of other workers and reports the value that leads to the maximum expected reward according to belief $Pr(x_p, x_q|x_w)$. In the final step, worker w is rewarded using a peer evaluation mechanism τ that uses the fact that workers who solve the same tasks have statistically more correlated answers than those who do not.

This setting depicts a typical crowdsourcing scenario: a group of workers is given a bundle of *a priori* similar tasks to solve, and each worker solves only some of the tasks from the bundle. For example, in text annotation, a requester (mechanism) could give 1000 sentences to annotate, and a group of, for example, 100 workers would be assigned to perform the tasks, in which each worker would annotate, for example, 50 sentences. Once the tasks are completed, the mechanism rewards workers for solving the tasks. To simplify the description of our mechanism, we will suppose that a worker

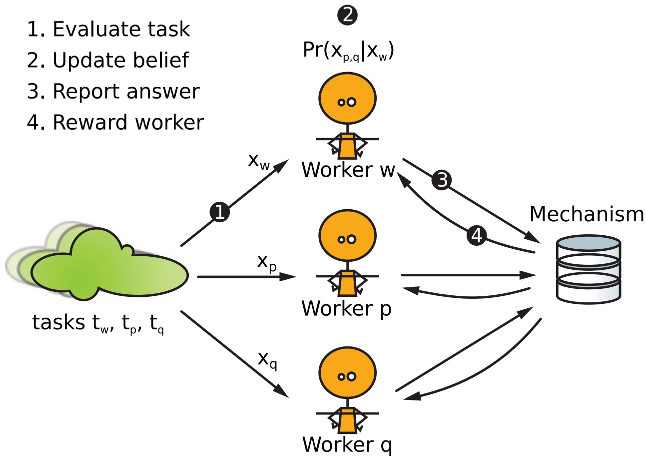

Fig. 1. The setting analyzed in this article.

w solves one task: this easily generalizes to cases in which worker w solves more than one task by applying the mechanism to each task solved by worker w separately.

The basic principle of a proper peer-prediction mechanism is to reward a respondent based on how *surprisingly common* that respondent's report is among the reports of peers. The PTS mechanism [Jurca and Faltings 2011; Faltings et al. 2014b] rewards a worker who reports answer x only if a randomly chosen peer who worked on the same task also gave the same answer. It uses a commonly known prior probability distribution $R(x)$ over possible answers x, and rewards a matching answer x with $\frac{1}{R(x)}$. It thus implements a principle of rewarding answers that are *surprisingly common*: common because they match those of a peer, and surprising because less likely answers carry a higher reward.

PTS has been shown to be useful in applications such as opinion polls [Garcin and Faltings 2014] and community sensing [Faltings et al. 2014a]. While PTS has been successfully applied in human computation [Faltings et al. 2014b], it has been limited to scenarios in which there is a known prior bias. For wider applications in crowdsourcing, there are two drawbacks:

ALGORITHM 1: The Peer Truth Serum for Crowdsourcing

Reward a worker w for solving task t_w as follows:

(1) Calculate the frequency of reported values within all tasks (excluding the report of worker w). Let us denote this frequency by R_w, which is equal to $R_w(x) = \frac{num(x)}{\sum_y num(y)}$, where *num* is the function that counts occurrences of reported values, and the summation in the denominator goes over all possible answers y.
(2) Select a peer worker p who was given task t_w to solve.
(3) Worker w is rewarded for reporting x_w with the score:

$$\tau(x_w, x_p) = \alpha \cdot (\tau_0(x_w, x_p) - 1), \tag{1}$$

where x_p is worker p's report, α is a constant strictly greater than 0, and τ_0 is defined as:

$$\tau_0(x_w, x_p) = \begin{cases} \frac{1}{R_w(x_w)} & \text{if } x_w = x_p \\ 0 & \text{if } x_w \neq x_p \end{cases}, \tag{2}$$

—The distribution R needs to be known.
—Consequently, workers can collude and report the least likely value $x_{min} = \arg\min_x R(x)$ to obtain significantly greater payoff than for honest reporting.

2.1. The Novel Mechanism

In this article, we show how to eliminate these drawbacks by obtaining R from the answer distribution of the workers themselves. The mechanism, called *Peer Truth Serum for Crowdsourcing* (PTSC), is shown in Algorithm 1. The mechanism represents a game in game theoretic context: the utility of each worker depends on reports (actions) of other workers. Therefore, we use an equilibrium analysis to determine the resulting behavior of workers.

The PTSC score requires only one peer p. If, however, we can assign multiple peers p to worker w (more than one peer worker p solves task t_w), the final score to worker w can be the average of the PTSC scores over all selected peers:

$$\tau(x_w, x_p) = \alpha \cdot \left(\frac{1}{n_{peers}} \sum_p \tau_0(x_w, x_p) - 1 \right) \tag{3}$$

where n_{peers} is the number of peers of worker w and τ_0 is defined by Equation (2). Scores (1) and (3) are equivalent in expectation; thus, the incentive properties of the mechanism are the same. However, using multiple reports reduces the variance in payments, which may often be desirable. Notice that the PTSC score (Equation (3)) is proportional to the ratio of the frequency of reports equal to x_w within task t_w and the frequency of reports equal to x_w within all the tasks (report x_w excluded). In other words, PTSC rewards worker w based on how surprisingly common worker w's report is: reports x_w that are more common within task t_w than expected by empirical frequency R receive higher rewards than those reports that are not as common as expected.[1]

We now give an intuitive understanding of the PTSC mechanism, assuming a scenario in which there are many tasks and the histogram of answers is thus a good approximation of the answer probability distribution. A more formal analysis, including cases with a small number of tasks, is given later in the article.

To decide on the best strategy, a worker w should estimate the reward expected for reporting answer x. This depends crucially on the worker's beliefs about the answer reported by the worker's peer. Notice that worker w does not need to know R. However, the worker's belief about $R(x)$ is equal to that worker's prior belief $P_p(x)$ about the evaluations of the other workers, provided that they are honest.

Consider first the case in which the worker does not solve the task at all. In this case, the worker should believe the peer answer to be distributed according to R used in Algorithm 1; thus, the expected reward is equal to zero for all possible reports. The worker thus has no interest in participating in the mechanism, and we can expect that such workers will not elect to participate.

Otherwise, the worker will have spent some effort to solve the task, and will endorse an answer x, that is, x is the worker's evaluation of the correct answer. Finally, the worker will form a posterior belief $P_{p|w}(x|x)$ about the answer given by the peer worker. We now come to a crucial assumption: the worker should believe that the answer of the peer will be positively correlated with the worker's answer. More precisely, the worker should believe that the posterior $P_{p|w}$ differs from the worker's prior P_p by giving the highest increase to the worker's own answer x:

$$\frac{P_{p|w}(x|x)}{P_p(x)} > \frac{P_{p|w}(\bar{x}|x)}{P_p(\bar{x})}, \forall \bar{x} \neq x \tag{4}$$

[1]To simplify our notation, we often omit subscript w from R_w.

To understand this condition, let us apply the Bayes' rule, which converts it to:

$$\frac{P_{w|p}(x|x)}{P_w(x)} > \frac{P_{w|p}(x|\bar{x})}{P_w(x)} \Leftrightarrow P_{w|p}(x|x) > P_{w|p}(x|\bar{x}), \forall \bar{x} \neq x,$$

or more elegantly $x = \arg\max_{\bar{x}} P_{w|p}(x|\bar{x})$. In other words, worker w believes in most likely endorsing answer x when the peer endorses the same answer x.

We discuss this condition in more detail later on; it is satisfied, for example, in Dirichlet-categorical models with Bayesian updating, while the simplest example in which the condition does not hold is when the answers of worker w and p are independent, that is, $P_{w|p}(x|\bar{x}) = P_w(x)$.

To see the importance of this condition, which we call the *self-predicting* condition, suppose that workers other than worker w are honest. In that case, the expected score of worker w for reporting y is $\alpha \cdot \frac{P_{p|w}(y|x)}{R(y)} - \alpha$, with $\frac{P_{p|w}(y|x)}{R(y)}$ being a good approximation of $\frac{P_{p|w}(y|x)}{P_w(y)}$. Thus, provided that worker w's beliefs satisfy the self-predicting condition, inequality (4) exactly shows that reporting x leads to the highest possible expected reward. Notice that the self-prediction differs from condition $x = \arg\max_{\bar{x}} P_{p|w}(\bar{x}|x)$, which states that worker w believes one's answer is adopted by the majority, and is arguably a stronger condition to satisfy. For example, this condition implies the self-prediction in binary answer spaces for a reasonable updating assumption $P_{p|w}(x|x) > P_p(x)$, while the other way around is not true.

Notice that worker w_1's and worker w_2's beliefs about the answers of other workers, and consequently their beliefs about R, need not be common. More precisely, each worker can have one's own private prior belief regarding what others report for a task that the worker has not solved, and one's own posterior belief regarding what others report for a task that the worker has solved. Furthermore, these beliefs incorporate the fact that workers can obtain their answers differently, for example, a worker might perceive oneself as more accurate than others. This is contrary to many existing mechanisms, in which either these beliefs are common among workers (agents) or are known to the center.

2.2. Example

To demonstrate the principles of the PTSC mechanism with parameter $\alpha = 1$, consider the following scenario. Let there be four possible answers $\{a, b, c, d\}$ and $n = 10$ tasks with the following correct answers and reports given by workers:

Task	Correct	Answers for the task
t_1	a	b, a, a, c
t_2	b	b, b, b, a
t_3	a	a, a, b, a
t_4	a	a, d, a, a
t_5	c	c, c, a, b
t_6	d	d, a, d, d
t_7	a	a, a, c, a
t_8	b	b, b, a, b
t_9	a	a, a, a, a
t_{10}	b	b, b, a, b

Each of the 10 tasks is solved by four different workers, giving us altogether 40 answers. For simplicity, we assume that each of the 40 answers is given by a different worker, that is, a worker solves only one task within the batch of 10 tasks. The collection of answers gives the following frequency of answers R:

Answer	a	b	c	d
Count	20	12	4	4
R	0.50	0.30	0.1	0.1

Now, consider an additional worker w who is given one of the tasks to solve, but whose answer is not in the earlier tables. The worker has a choice between three actions:

—*Heuristic*: Invest no effort and choose an answer that is independent of the task.
—*Honest*: Invest high effort to find the answer and report it truthfully.
—*Strategic*: Invest high effort to find the answer, but report an answer that may not be truthful.

As a rational agent, worker w will choose the action that maximizes the expected reward, based on what the worker believes about the answers of other workers. In particular, the expected reward depends on the probability of matching the peer answer.

For the *heuristic* action, the answer x is independent of the task; thus, the probability of having a matching peer answer is equal to the frequency of this answer among all tasks, $R(x)$. Therefore, the expected reward is:

$$R(x) \cdot \left(\frac{1}{R(x)} - 1 \right) + (1 - R(x)) \cdot (-1) = 0$$

no matter what the answer x is, or what the worker believes about $R(x)$. This also means that any strategy that chooses the answer independently of the specific task will carry an expected reward of 0.

For the other two actions, the worker solves the task and we assume that the worker has found an answer x. If the worker believes that the peers are likely to find the same answer and report it truthfully, the worker should believe that answer x is not less likely for this task than in the distribution over all tasks. In particular, the worker should believe that answer x is the one with the highest increase, as given in the self-predicting condition (Equation (4)). In the example, the answer distributions for the tasks, grouped by their correct answers, are as follows:

		Observed answer			
Correct		a	b	c	d
a	$Count(a)$	15	2	2	1
	$freq(\cdot\|a)$	0.75	0.1	0.1	0.05
b	$Count(b)$	3	9	0	0
	$freq(\cdot\|b)$	0.25	0.75	0	0
c	$Count(c)$	1	1	2	0
	$freq(\cdot\|c)$	0.25	0.25	0.5	0
d	$Count(d)$	1	0	0	3
	$freq(\cdot\|d)$	0.25	0	0	0.75

We can see that, for each group of tasks, it satisfies the self-predicting condition (Equation (4)).

Let us assume thus that worker w has found answer a and let worker w's beliefs about the peers' answers, for simplicity, be as shown in the table, that is, $P_{p|w}(\cdot|a) = freq(\cdot|a)$. It is important to note that worker w does not know the answers of other workers, thus has to reason about them by taking the expectation over all possibilities. The values in the table are based on the realization of one possibility. However, they satisfy the self-predicting condition; thus, they represent one candidate for a worker's belief.

The reasoning of worker w for the task would be as follows. For reporting the worker's true answer a, the probability of matching the peer is 0.75, so worker w expects to get a payoff equal to $\frac{0.75}{0.5} - 1 = 0.5$ for the *honest* action. This is greater than what the worker expects to get with the *heuristic* action (0), or the *strategic* action with answer b ($\frac{0.1}{0.3} - 1 = -2/3$), c ($\frac{0.1}{0.1} - 1 = 0$), or d ($\frac{0.05}{0.1} - 1 = -0.5$).

We will show in Sections 4.2 and 4.3 that this holds not only for accurate beliefs, but for any belief about peer answers that satisfies the self-predicting condition (Equation (4)). Just like constant rewards, the payment should be scaled so that the expected reward exceeds the cost of effort invested in solving the task. We will address this question in Section 4.6.

Now, consider what happens when workers collude, so that, for example, those with evaluations a and d report a, while those with evaluations b and c report b. In this case, a worker w with evaluation a believes that the probability of a peer reporting a is $P_{p|w}(a|a) + P_{p|w}(d|a)$. However, R also has different values than for the honest strategy profile; $R(a)$ for the colluding strategy profile is equal to $R(a) + R(d)$ for the honest strategy profile, and exactly compensates for the gain in matching probability. In this example, worker w's expected payoff is $\frac{0.75+0.05}{0.5+0.1} - 1 = 0.33$, and is less than what worker w gets when everybody (including this worker) reports honestly. Thus, the collusion is not profitable; we show that this holds, in general, in Section 4.4.

Finally, the mechanism is robust against peers that provide low-quality results, as is common in crowdsourcing. Assume that 10% of workers report honestly, while others report randomly with probability of reporting a equal to 30%. Then, worker w, whose evaluation is a, believes that the probability of a peer reporting a is $0.1 \cdot P_{p|w}(a|a) + 0.9 \cdot 0.3$, while $R(a)$ is, in this case, equal to $0.1 \cdot R(a) + 0.9 \cdot 0.3$, where the latter $R(a)$ is the one calculated for the honest strategy profile. Hence, the expected payoff of worker w for reporting a is $\frac{0.1 \cdot 0.75 + 0.9 \cdot 0.3}{0.1 \cdot 0.5 + 0.9 \cdot 0.3} - 1 = \frac{0.345}{0.32} - 1$, which is more than worker w expects to obtain for random reporting. With the appropriately scaled payoff, the same holds for worker w's profit. Thus, even a small fraction of cooperative workers suffices to create the right incentives. We will show this more formally in Section 4.3.

The important condition for the discussed collusion-resistant properties to hold is that workers distinguish tasks only by their endorsed answers (evaluations). If tasks were distinguished by other features, workers could use those features to form a colluding strategy that is more profitable than honest reporting. Thus, we need to be careful to apply the mechanism to a batch of tasks that are, on the surface, very similar. Fortunately, most crowdsourcing tasks satisfy this condition.

2.3. Application Examples

The scenario depicted by Figure 1 captures many interesting crowdsourcing tasks. These include objective tasks that have correct answers and subjective tasks for which workers are asked to provide their opinions. We present two examples of such crowdsourcing tasks, community (participatory) sensing and peer grading, which we use to evaluate the quality of the PTSC mechanism.

Community Sensing. In a typical community (participatory) sensing scenario, a group of sensors measure a physical phenomenon. That is, private mobile devices equipped with sensors acquire information about a spatially distributed phenomenon, such as air pollution or weather. Since sensing induces a cost due to the fact that sensing modules need to be installed and maintained, the party interested in monitoring the physical phenomenon needs to incentivize the crowd to incur this cost and provide quality data.

A peculiar property of a community sensing setting is that a mechanism has no control over sensing devices, nor does it have a way of directly verifying the correctness of the obtained data. This leads us to peer consistency mechanisms. One of the peer consistency methods proposed for information elicitation in the community sensing setting is the mechanism from Faltings et al. [2014a]. The major drawback of this mechanism is that noninformed reporting strategies (strategies for which sensors do not make measurements) can result in significantly higher expected payoffs than honest reporting. We show both theoretically and experimentally that PTSC solves this problem. In this context, PTSC extends the mechanism from Radanovic and Faltings [2015a] by allowing a less dense sensor network.

Peer Grading. One of the main challenges in massive open online courses (MOOCs) is evaluation of student assignments. This is especially true if assignments are essay questions that cannot be graded automatically. In such cases, peer-grading techniques can be applied: participants (students) grade assignments of their colleagues, and the grade of each student is obtained by aggregating the peer grades.

Peer grading in MOOCs represents a typical crowdsourcing scenario, in which workers are students who are assigned to grade their own assignments. A proper monitoring of such a grading system is not feasible due to the number of participants; thus, quality control has to be designed in the form of incentives. Moreover, the incentives have to take into account that participants have different grading abilities and are inclined to manipulate the reward system.

Often, the quality control in subjective tasks is achieved by using a peer consistency mechanism that rewards workers when their reports agree [Huang and Fu 2013b]. This type of mechanism, however, does not take into account that workers may have a potential bias toward more likely evaluations. That is, workers who believe that their opinion is not the most common are incentivized to misreport. Moreover, colluding strategies in which workers report the same value result in higher payoffs; such behavior is likely to occur [Gao et al. 2013].

We propose the PTSC as a suitable mechanism for the two crowdsourcing scenarios due to its strong incentive properties: its ability to cope with collusive behavior while making honest reporting the most profitable equilibrium.

3. FORMAL SETTING

Our crowdsourcing model is similar to the one presented in Dasgupta and Ghosh [2013], with the basic structure depicted by Figure 1. We formalize it in the following way.

3.1. Tasks

We consider a crowdsourcing scenario in which a group of workers solves n statistically independent tasks from a set of *a priori* similar tasks $T = \{t_1, t_2, \ldots\}$, and are rewarded based on their performance. Tasks are *a prior* similar if they are only distinguished by their correct answers; examples are collecting sensor measurements at different locations, grading student answers to the same question, or interpreting similar images. We thus identify a task t with its correct answer $X^t_{correct}$ that is generated randomly according to a distribution function defined over a discrete and finite answer space $\mathcal{X} = \{x, y, z, \ldots\}$. Note that the existence of a correct answer does not necessarily restrict applicability of the proposed approach to eliciting objective information. Models of obtaining (subjective) preferences over alternatives often assume that preferences are noisy evaluations of the true state of the world (e.g., the random utility model [Soufiani et al. 2012]). The only important consideration that needs to be taken into account is that a correct answer $X^t_{correct}$ is not accessible to a mechanism, while the answers of different workers to the same task should have some minimal correlation (in our case, expressed by the self-predicting condition).

Each task in set T is answered by at least 2 different workers, randomly chosen from a large pool $W = \{w, p, q, \ldots\}$ of available workers. Without loss of generality, we can assume that a worker w solves only one task t_w in a family of tasks T. If this assumption does not hold within set T, we simply partition T into subsets that satisfy the assumption, and apply a reward mechanism to each subset separately. The overall reward of a worker w can then be defined as an average or a sum of the obtained rewards.

3.2. Workers

A worker in W is assumed to be a risk-neutral rational agent who aims to maximize expected profit. With that in mind, our crowdsourcing setting can be viewed as a two-stage game, in which in the first-stage workers choose the amount of effort they want to invest in solving their task, and in the second stage, they decide on what to report.

When a worker w solves a task, that worker invests a certain amount of effort e_w. We assume two levels of effort, *high* and *low*; low effort, denoted as e_0, should intuitively be seen as a random answer, provided automatically, without understanding the task; high effort (e_1), on the other hand, is the work exerted by an honest worker who does one's reasonable best to answer the task correctly. When the rewards offered by the requester are insufficient to cover the cost of high effort, workers may choose an approximate strategy whose effort falls in between these extremes. To simplify the analysis, we would consider this intended effort the high-effort case, thus consider only strategies with either low or high effort. Unlike accuracy, workers' utility decreases as their effort increases, meaning that a worker w experiences cost $c_w(e_w)$ for investing effort e_w—cost c_w is an increasing function of effort e_w, that is, $c_w(e_0) < c_w(e_1)$, and can be different for different workers.

In many real scenarios, tasks can be solved with multiple levels of effort according to a possibly very complex cost–benefit relation. We do not consider the question of designing mechanisms to encourage intermediate levels of effort to optimize cost-effectiveness. We note that, by scaling the reward function—in PTSC, increasing parameter α—it is possible to incentivize higher levels of effort.

When a worker solves a task, that worker obtains an *evaluation* of the correct answer to the task, which we model as a random variable X_w that takes values from the set of possible answers $\mathcal{X}$. Note that worker w's evaluation X_w does not have to be equal to the correct answer $X^{t_w}_{correct}$. In order to formally define workers' action space, we extend answer space $\mathcal{X}$ by adding the symbol $\emptyset$, meaning that a worker w whose evaluation X_w is equal to $\emptyset$ has not solved the task (i.e., $e_w = e_0$).

Once worker w solves the task, that worker reports an answer to the mechanism. Reported value Y_w can differ from evaluation X_w, either because worker w lies or because worker w does not solve the task ($X_w = \emptyset$). We see that a worker w faces a choice between three basic strategies:

—*Honest*: Invest high effort e_1 to obtain evaluation $X_w = x$, and report honestly $Y_w = x$.
—*Strategic*: Invest high effort e_1 to obtain evaluation $X_w = x$, but report $Y_w = y$ that is randomly generated according to a distribution $Q_{w|w}(y|x)$. Naturally, *honest* strategy is equal to *strategic* strategy for $Q_{w|w}(y|x) = \mathbb{1}_{y=x}$, where $\mathbb{1}_{cond}$ is an indicator variable equal to 1 when condition *cond* is satisfied, and otherwise is 0.
—*Heuristic*: Invest low effort e_0 (i.e., $X_w = \emptyset$) and report according to a distribution $Q_{w|w}(y|\emptyset)$. Note that this strategy includes both random reporting and heuristic reporting in which workers agree which values to report in advance (e.g., they all report the same value).

All the three strategies can be described using probability distribution function $Q_{w|w}$. Intuitively, workers' strategies are either based on their evaluations or are heuristic (random) if workers choose not to invest high effort in solving tasks. In principle, each worker can have one's own strategy. However, because there is a large pool of workers and workers are randomly assigned to tasks, we can restrict our attention to symmetric strategies and abuse our notation by denoting symmetric strategy profiles with *honest*, *strategic*, and *heuristic*. Namely, from a worker w's perspective, asymmetric strategies of other workers q can be seen as symmetric strategies with $Q_{q|q}$ obtained by averaging workers q's strategies. Hence, *honest* strategy profile has effectively the same properties even when workers are allowed to have asymmetric strategies.

3.3. Workers' Beliefs

Workers' beliefs are characterized by two distributions that need not be the same for different workers. A prior belief of a worker w is a probability distribution function $P_q(x) = Pr(X_q = x)$ regarding the evaluation X_q of a randomly chosen worker q who has solved an arbitrary task in T. A posterior belief of a worker w is a probability distribution $P_{p|w}(x|y) = Pr(X_p = x|X_w = y)$ regarding the evaluation X_p of a worker p who has solved the same task. In particular, when worker w solves a task, that worker also gains an insight regarding the evaluation of a worker p; thus, worker w updates a prior belief $P_p(x)$ to obtain a posterior belief $P_{p|w}(x|y) = Pr(X_p = x|X_w = y)$.

The consequence of tasks in T being statistically independent implies that $P_{q|w}(x|y) = P_q(x)$ for a worker q who has solved a different task than worker w. We call workers p *peers*, and workers q *reference* workers of worker w. Because tasks are randomly distributed to workers from a large pool W,[2] prior belief regarding peers' evaluations is equal to prior belief regarding the evaluations of reference workers, that is, $P_p(x) = P_q(x)$. Moreover, when $X_w = \emptyset$ (i.e., $e_w = 0$), worker w gains no insight into peers' evaluations; thus, $P_{p|w}(x|\emptyset) = P_p(x)$.

We assume workers' beliefs to be *fully mixed*, meaning that for any two workers w and p, we have that $\forall x, y \in \mathcal{X} : P_p(x) > 0, P_{p|w}(x|y) > 0$. This is a common assumption in many similar settings (e.g., Miller et al. [2005] and Prelec [2004]); it states that workers believe that all answers can be endorsed by their potential peers.

The expected payoff of a worker w depends on the reports of peers and reference workers rather than on their evaluations. Therefore, worker w's beliefs regarding evaluations (P_p, $P_{p|w}$, and P_q), need to be transformed into the beliefs regarding other workers' reports, here denoted by Q_p, $Q_{p|w}$, and Q_q. Using the fact that $Q_{p|p}$ defines worker p's strategy, we obtain that the proper transformations for a *strategic* strategy profile are $Q_q(y) = Q_p(y) = \sum_{z\in\mathcal{X}} Q_{p|p}(y|z)P_p(z)$ and $Q_{p|w}(y|x) = \sum_{z\in\mathcal{X}} Q_{p|p}(y|z)P_{p|w}(z|x)$. When workers use the *honest* strategy profile, Q_q, Q_p, and $Q_{p|w}(y|x)$ reduce to $Q_q = Q_p = P_p$ and $Q_{p|w} = P_{p|w}$. For a *heuristic* strategy profile, we obtain $Q_q = Q_p = Q_{p|w} = Q_{p|p}$.

Since rewards are based on the comparison of reports, some minimal correlation between workers' evaluations must exist. We incorporate this correlation in the workers' belief systems by assuming the *self-predicting* condition [Jurca and Faltings 2011; Radanovic and Faltings 2013]:

Definition 3.1. Consider workers w and p who solve the same task. A worker w has a belief system that satisfies the *self-predicting* condition iff[3]

$$\frac{P_{p|w}(y|x)}{P_p(y)} - 1 < \frac{P_{p|w}(x|x)}{P_p(x)} - 1, \forall y \neq x.$$

[2]This translates to having workers whose *proficiencies* (*qualities*) are i.i.d. random variables, in which the proficiency of a worker might be known to that worker, but not to the other workers. Proficiency can be thought of as a probability of obtaining the correct evaluation.

[3]We keep -1 on both sides to make the proofs and the notion of the *self-predictor* clear.

Moreover, for worker w, whose beliefs satisfy the self-predicting condition, we define *self-predictor* Δ_w as the smallest number in [0, 1] so that

$$\frac{P_{p|w}(y|x)}{P_p(y)} - 1 < \left(\frac{P_{p|w}(x|x)}{P_p(x)} - 1\right) \cdot \Delta_w, \forall y \neq x \tag{5}$$

holds.

The self-prediction holds in the common case in which a worker believes that only the evaluation one is endorsing is more likely than in the prior distribution, as all other evaluations would become less likely. This also includes binary answer spaces (as in Dasgupta and Ghosh [2013]), as well as a more general case when workers observe different samples drawn from the same categorical distribution, but with unknown parameters sampled from a Dirichlet distribution. If a worker w uses Bayesian updating, that worker's posterior $P_{p|w}$ is greater than the prior P_p only for the worker's own evaluation, implying the self-predicting condition. Note that, in this case, $P_{p|w}(x|x)$ can be less than $P_{p|w}(y|x)$ for $y \neq x$. In the following 2 scenarios, the self-prediction possibly holds, although a worker might believe that others may answer differently:

—when a worker's evaluation is *a priori* unlikely, such as receiving bad service from a hotel that is otherwise very highly rated, and
—when there are strongly correlated values, for example, when measuring a temperature of 26 degrees, answers of 25 and 27 degrees will also be very likely.

We characterize the degree of correlation that a worker w believes to be possible by the *self-predictor* Δ_w: the smaller it is, the more proficient workers are, that is, different answer values are less correlated. For example, $\Delta_w = 0$ indicates that different answer values are not correlated, while for $\Delta_w \approx 1$ workers are more likely to confuse two similar answers.

To model possible differences in workers' beliefs, we assign to each worker w a belief type $\theta_{P,w}$. Belief type $\theta_{P,w}$ is an element of an abstract set Θ_P; it determines how beliefs P_p and $P_{p|w}$ are formed. We call belief type $\theta_{P,w}$ of a worker w *admissible* if the associated beliefs P_p and $P_{p|w}$ comply with the conditions described in this section (the self-predicting condition included).

3.4. Subjective Equilibrium

As a rational agent, a worker aims to maximize profit; in case of uncertainties, the worker is assumed to maximize *expected* profit. Since a worker's payoff depends on what other workers report, we use an equilibrium analysis to determine the resulting behavior of workers. In particular, a strategy profile $\sigma = (\sigma_w, \sigma_p, \sigma_q, \ldots)$, which represents a collection of strategies of workers $\{w, p, q, \ldots\}$, is an *equilibrium* if for any worker $\bar{w} \in \{w, p, q, \ldots\}$, the worker's expected profit is maximized when adopting strategy $\sigma_{\bar{w}}$, that is, $\sigma_{\bar{w}}$ is the best response. An equilibrium is *strict* if any other strategy $\sigma'_{\bar{w}} \neq \sigma_{\bar{w}}$ leads to a strictly lower expected profit. If a mechanism admits *honest* reporting as an equilibrium, we say that it is *incentive compatible*.

Since beliefs need not be common among workers, that is, they are *subjective*, we are particularly interested in an equilibrium concept called *ex-post subjective equilibrium* (Witkowski and Parkes [2012a]), which is defined over *admissible belief types*. In this equilibrium concept, a worker's best response is independent of the belief types of other workers. The crowdsourcing model has the form of a two-stage game; thus, in our analysis we use a refinement of an ex-post subjective equilibrium that we call *perfect ex-post subjective equilibrium*. Here, *perfect* means that a worker chooses at each stage a strategy that is in expectation the best response to the strategies of other workers according to one's current beliefs. In our case, a worker would first calculate whether, according to one's prior belief, one can expect to profit from investing high effort. If profit can be expected, the worker would solve the task, update one's belief with respect to the evaluation, and choose the best answer to report. If not, the worker would simply report the best answer according to prior belief. For simplicity, we omit the full name of the equilibrium concept in the remaining part of the article.

4. ANALYZING THE PEER TRUTH SERUM FOR CROWDSOURCING

In Section 2, we introduced the PTSC mechanism with the reward function:

$$\tau(x_w, x_p) = \alpha \cdot \left(\tau_0(x_w, x_p) - 1\right)$$
$$\tau_0(x_w, x_p) = \begin{cases} \frac{1}{R_w(x_w)} & \text{if } x_w = x_p \\ 0 & \text{if } x_w \neq x_p \end{cases},$$

which is applicable when a large quantity of statistically independent tasks can be used for constructing R. Note that the PTSC score $\tau(x_w, x_p)$ might take negative values. To avoid negative rewards, one can additionally reward workers with α. However, we keep the form (1) to relate PTSC to the mechanism that elicits proper values of parameter α, presented later in the article.

The key concept for analyzing PTSC in game theoretic terms is workers' expected reward. Suppose that worker w believes that the other workers are honest. In that case, worker w expects that the frequency of reports equal to y within the task t_w is $P_{p|w}(y|x)$, where the evaluation of worker w is x (see Section 3.3). On the other hand, the expected frequency of reports equal to y within a task $t \neq t_w$ not solved by worker w is $P_q(y) = P_p(y)$. Therefore, for a large set of tasks, the expected score of worker w for reporting y is approximately equal to

$$\alpha \cdot \left(\frac{P_{p|w}(y|x)}{P_p(y)} - 1 \right).$$

The claim follows from the law of large numbers and the statistical independence of tasks. Similarly, when workers adopt a strategy described by a distribution $Q_{p|p}$ (*strategic* or *heuristic* strategies), we obtain that worker w's belief regarding the frequency of reports equal to y within the task t_w is $Q_{p|w}(y|x) = \sum_{z\in\mathcal{X}} Q_{p|p}(y|z)P_{p|w}(z|x)$. Worker w's belief regarding the frequency of reports equal to y within a task $t \neq t_w$ not solved by worker w is $Q_q(y) = Q_p(y) = \sum_{z\in\mathcal{X}} Q_{p|p}(y|z)P_p(z)$. Thus, the expected score of worker w for reporting y is approximately equal to

$$\alpha \cdot \left(\frac{Q_{p|w}(y|x)}{Q_p(y)} - 1 \right)$$

when $Q_p(y) > 0$ and $-\alpha$ when $Q_p(y) = 0$, as, in that case, a peer does not report y.

By analyzing the structure of workers' expected scores for different strategies, we can deduce several properties of the PTSC mechanism:

—The expected payoff of a worker w who invests high effort, obtains evaluation x, and who believes that the other workers are honest, for reporting x is $\alpha \cdot (\frac{P_{p|w}(x|x)}{P_p(x)} - 1)$. Due to the self-predicting condition (even with the self-predictor $\Delta_w \approx 1$), this is greater than what the worker expects to obtain for reporting $y \neq x$:

$$\alpha \cdot \left(\frac{P_{p|w}(x|x)}{P_p(x)} - 1 \right) > \alpha \cdot \left(\frac{P_{p|w}(y|x)}{P_p(y)} - 1 \right).$$

Moreover, the *honest* reporting strategy leads to a strictly positive expected payoff because $\frac{P_{p|w}(x|x)}{P_p(x)} > 1$. By choosing an appropriate scaling parameter α, one can cover the cost of effort $c(e_1)$, making the payment scheme ex-ante individually rational.

—When workers adopt a *heuristic* strategy profile (investing low effort), their expected payoff for reporting y is equal to 0, because $Q_{p|w} = Q_p$ (see Section 3.3). This is in contrast to the *honest* reporting strategy profile in which the expected payoff is greater than 0. Therefore, by appropriately scaling parameter α, one can overcome the problem of having the higher cost for investing effort ($c(e_1) > c(e_0)$) and incentivize workers to invest effort. This further implies that *honest* reporting is an equilibrium strategy.

—Finally, we can analyze if workers can manipulate the mechanism in order to obtain higher payoffs. Suppose workers adopt a *strategic* strategy profile, that is, each worker p obtains a private evaluation x_p, but reports according to a distribution $Q_{p|p}(\cdot|x_p)$. In that case, the expected payoff of worker w whose evaluation is x for reporting y, is equal to

$$\alpha \cdot \left(\frac{Q_{p|w}(y|x)}{Q_p(y)} - 1 \right) = \alpha \cdot \left(\frac{\sum_z Q_{p|p}(y|z) \cdot P_{p|w}(z|x)}{\sum_z Q_{p|p}(y|z) \cdot P_p(z)} - 1 \right)$$

when $Q_{p|p}(y|z) > 0$ for some $z \in \mathcal{X}$. This follows from the definitions of $Q_{p|w}$ and Q_p. Assuming the self-predicting condition (even with the self-predictor $\Delta_w \approx 1$), the expression is maximized when $Q_{p|p}(\bar{z}|z) = \mathbb{1}_{\hat{\sigma}(z)=\bar{z}}$ with $\hat{\sigma}(x) = y$, where $\hat{\sigma}$ is a bijective function from the set of possible evaluations to the set of possible reports. Namely, the earlier expression can be written as

$$\begin{aligned}
&\alpha \cdot \left(\frac{Q_{p|p}(y|x) \cdot P_{p|w}(x|x) + \sum_{z \neq x} Q_{p|p}(y|z) \cdot P_{p|w}(z|x)}{Q_{p|p}(y|x) \cdot P_p(x) + \sum_{z \neq x} Q_{p|p}(y|z) \cdot P_p(z)} - 1 \right) \\
&= \alpha \cdot \left(\frac{Q_{p|p}(y|x) \cdot P_p(x) \cdot \frac{P_{p|w}(x|x)}{P_p(x)} + \sum_{z \neq x} Q_{p|p}(y|z) \cdot \frac{P_{p|w}(z|x)}{P_p(z)} \cdot P_p(z)}{Q_{p|p}(y|x) \cdot P_p(x) + \sum_{z \neq x} Q_{p|p}(y|z) \cdot P_p(z)} - 1 \right) \\
&\leq \alpha \cdot \left(\frac{Q_{p|p}(y|x) \cdot P_p(x) \cdot \frac{P_{p|w}(x|x)}{P_p(x)} + \sum_{z \neq x} Q_{p|p}(y|z) \cdot \frac{P_{p|w}(x|x)}{P_p(x)} \cdot P_p(z)}{Q_{p|p}(y|x) \cdot P_p(x) + \sum_{z \neq x} Q_{p|p}(y|z) \cdot P_p(z)} - 1 \right) \\
&= \alpha \cdot \left(\frac{P_{p|w}(x|x)}{P_p(x)} - 1 \right),
\end{aligned}$$

where the inequality comes from the self-predicting condition and is strict if there exists $z \neq x$ such that $Q_{p|p}(y|z) > 0$. Since the *honest* reporting strategy profile can be described by $\hat{\sigma}(x) = x$ and results in a payoff equal to $\alpha \cdot (\frac{P_{p|w}(x|x)}{P_p(x)} - 1)$, we conclude that the maximum payoff is obtained for honest reporting. The same holds for profit if α is properly chosen.

The PTSC mechanism with reward function (1) assumes a large number of statistically similar tasks, which is often satisfied in practice. In the following sections, we adopt this approach to construct a more robust version of the PTSC mechanism, which also operates with a smaller number of statistically independent tasks. Therefore, we provide a formal analysis only for the more general version of the PTSC mechanism.

ALGORITHM 2: The Robust Peer Truth Serum for Crowdsourcing

Reward each worker w using the following mechanism:

(1) Consider $n-1$ tasks in addition to task t_w, where n is big enough to allow desirable properties (see Theorem 4.3, Theorem 4.7, and Section 4.5).
(2) Randomly sample n reports from n different tasks, including the task t_w, but not worker w's report.
(3) Calculate the frequency of reported values within this sample $R_w(x) = \frac{num(x)}{\sum_{y \in \mathcal{X}} num(y)}$, where *num* is the function that counts occurrences of reported values in the sample.
(4) Worker w is rewarded for reporting $Y_w = x_w$ with the score

$$\tau(x_w, x_p) = \begin{cases} \alpha \cdot \left(\frac{\mathbb{1}_{x_w = x_p}}{R_w(x_w)} - 1 \right) & \text{if } R_w(x_w) \neq 0 \\ 0 & \text{if } R_w(x_w) = 0 \end{cases}, \qquad (6)$$

where $\mathbb{1}_{x_w = x_p}$ is an indicator variable (equal to 1 if $x_w = x_p$, and 0 otherwise) and α is a constant strictly greater than 0. x_p is the report of worker w's peer, who solves task t_w and whose report is in the sample from which R_w was obtained.

4.1. Robust Peer Truth Serum for Crowdsourcing

In the PTSC mechanism from the previous section, reports from all workers were used to calculate statistic R. In general, this might not lead to a proper result if one task is solved by significantly more workers than other tasks. To avoid this issue, we sample an answer from each task and construct R from the obtained sample.

Since R is calculated from a finite number of samples, it is possible that $R(x)$ is equal to 0 for a certain report x, which would lead to an ill-defined score due to the division by 0. To overcome this problem, we distinguish values x when the statistic $R(x)$ is equal to 0. When $R(x) \neq 0$, a worker who reports x obtains a score proportional to $\frac{1}{R(x)} - 1$ if one's peer has also reported x, and a score proportional to -1 otherwise. On the other hand, if $R(x) = 0$, a worker who reports x obtains 0, since there is no peer that matches the worker's report. As we will see later, this definition ensures us that workers who invest low effort are indifferent between reporting any two different answers, which is important when it comes to suppressing heuristic reporting strategies.

More formally, we define the Robust Peer Truth Serum for Crowdsourcing (RPTSC) as a mechanism that follows the steps of Algorithm 2.

Although RPTSC is a nonlinear scheme, the expected score can be expressed in a closed form.

LEMMA 4.1. *The expected payoff of worker w with evaluation $X_w = x$ and report $Y_w = y$ in the RPTSC mechanism is equal to*

$$\begin{cases} \alpha \cdot \left(\frac{Q_{p|w}(y|x)}{Q_p(y)} - 1 \right) \cdot (1 - (1 - Q_p(y))^{n-1}) & \text{if } Q_p(y) > 0 \\ 0 & \text{if } Q_p(y) = 0 \end{cases}, \qquad (7)$$

where n is the number of tasks.

An important property to satisfy is a resilience toward heuristic (random) reporting, meaning that, in expectation, a mechanism should not reward workers who invest low effort, regardless of their reporting strategy. The following proposition shows that RPTSC satisfies this property.

PROPOSITION 4.2. *In the RPTSC mechanism, the expected payoff of a worker w with a heuristic strategy is equal to 0 and that worker's profit is equal to $-c_w(e_0)$.*

In order to incentivize workers to invest high effort, their expected payoff for investing high effort should be strictly greater than when they report randomly. Suppose that all workers adopt the *honest* strategy. We denote by $\bar{\tau}_w(\alpha)$ the expected payoff of a worker w before the worker evaluates a task, that is:

$$\bar{\tau}_w(\alpha) = \mathbb{E}_{X_w=x}\left(\alpha \cdot \left(\frac{P_{p|w}(x|x)}{P_p(x)} - 1\right) \cdot \left(1 - (1 - P_p(x))^{n-1}\right)\right), \tag{8}$$

where the expectation $E_{X_w=x}$ is taken over worker w's possible evaluations $x \in \mathcal{X}$. It is important to note that, here, all workers, including worker w, adopt a strategy of investing high effort e_1 and truthful reporting. This differs from random reporting strategies, in which workers invest low effort e_0 and are expected to obtain payoffs equal to 0. Also, note that $\bar{\tau}_w(\alpha)$, which represents worker w's expected payoff prior to evaluation, depends on parameter α.

4.2. Incentive Compatibility

When workers agree to invest high effort e_1 and report honestly, a worker w might find it more profitable to deviate, as investing high effort increases the cost. In order to prevent such deviations, a mechanism should cover the cost of investing high effort; in RPTSC, this can be done by properly scaling parameter α.

THEOREM 4.3. *Suppose that for all workers w and answers $x \in \mathcal{X}$, parameter α and the number of tasks n satisfy*

$$\bar{\tau}_w(\alpha) > c_w(e_1) - c_w(e_0)$$
$$\frac{1 - (1 - P_q(x))^{n-1}}{1 - P_q(x)^{n-1}} \geq \Delta_w \tag{9}$$

where Δ_w is the self-predictor of worker w. Then, the RPTSC mechanism admits the honest reporting strategy profile as a strict equilibrium.

It is interesting to note that reporting according to a strategy profile *strategic* defined by $Q_{w|w}(y|x) = \mathbb{1}_{y=\hat{\sigma}(x)}$, where $\hat{\sigma}$ is a bijective function $\hat{\sigma} : \mathcal{X} \rightarrow \mathcal{X}$ from the answer space to the set of reports, is also an equilibrium, provided that the conditions of Theorem 4.3 hold. For example, bijection $\hat{\sigma}$ could define that workers with evaluation x report y, while those with evaluation y report x. This symmetry comes from Lemma 4.1 and the fact that $Q_p(\hat{\sigma}(y)) = P_p(y)$ and $Q_{p|w}(\hat{\sigma}(y)|x) = P_{p|w}(y|x)$. We call this property *permutation indifference*, which implies that the equilibrium strategies defined by $Q_{w|w}(y|x) = \mathbb{1}_{y=\hat{\sigma}(x)}$ achieve the same expected profits as the *honest* strategy profile. However, these strategies require costly coordination without any benefit.

4.3. Low-Effort Aversion

Although RPTSC admits noninformed equilibria, Proposition 4.2 shows that workers are not expected to profit in these equilibria. In fact, the direct consequence of Proposition 4.2 and Theorem 4.3 is that a low-effort strategy profile results in lower expected profits than the *honest* strategy profile for an appropriately scaled parameter α, even if the low-effort strategy is a mixture of investing high and low effort.

LEMMA 4.4. *Suppose that Condition* (9) *of Theorem 4.3 holds. Any equilibrium of RPTSC in which a heuristic strategy is adopted (played) with nonzero probability, that is, where a worker w's effort e_w can be equal to $e_w = e_0$, leads to lower expected profits than the honest reporting strategy profile.*

With RPTSC, one can achieve even more. Suppose that a population of workers contains a certain number of workers who invest high effort and report their true evaluations, and other workers whose strategies are based on low effort. Then, RPTSC can be properly scaled so that the best response to this scenario is to invest high effort. This means that a fraction of at least β honest workers can be used to eliminate low-effort equilibria. We call this property *low-effort* aversion.

Definition 4.5. Consider a parameter $\beta \in (0, 1]$ and a strategy profile that is a mixture of the *honest* and *heuristic* strategies, in which the *honest* strategy is adopted (played) with probability γ. A mechanism is *β-low-effort* averse if it does not admit the mixed-strategy profile as an equilibrium for any γ such that $\beta \leq \gamma \leq 1$.

PROPOSITION 4.6. *Suppose that scaling parameter α is such that*

$$\alpha > \frac{c_w(e_1) - c_w(e_0)}{\beta \cdot \mathbb{E}_{X_w=x}[P_{p|w}(x|x) - P_p(x)]} \tag{10}$$

for all workers w, where $\mathbb{E}_{X_w=x}$ is the expectation over possible evaluations of a worker w. Then, RPTSC is β-low-effort averse.

4.4. Optimality

Many payment schemes are not resistant to collusive strategies in which all workers report identical values; often, these strategies lead to considerably greater payoffs than the honest-reporting strategy profile. One can easily show by using Lemma 4.1 that the expected payoff of a strategy profile in which workers report a single value is equal to 0 in the RPTSC mechanism. Since it is more profitable for workers to invest high effort and report honestly, we can say that appropriately scaled RPTSC is resistant to collusion based on reporting a single value. However, it is reasonable to ask how good the honest-reporting equilibrium is compared to the optimal strategy profile. The following theorem gives a condition for the optimality of honest reporting.

THEOREM 4.7. *Suppose that for all workers w and answers $x \in \mathcal{X}$, parameter α and the number of tasks n satisfy*

$$\begin{gathered} \bar{\tau}_w(\alpha) > c_w(e_1) - c_w(e_0) \\ \left(1 - (n-1) \cdot P_p(x) \cdot \frac{(1 - P_p(x))^{n-2}}{1 - (1 - P_p(x))^{n-1}}\right) \geq \Delta_w \end{gathered}, \tag{11}$$

where Δ_w is the self-predictor of worker w. Then, the honest reporting strategy profile is a strict equilibrium *of the RPTSC mechanism that results in the maximum profit.*

Condition (11) is stricter than Condition (9) in a sense that any self-predictor Δ_w that satisfies Condition (11) necessarily satisfies Condition (9).

LEMMA 4.8. *If for all workers w and answers $x \in \mathcal{X}$, self-predictor Δ_w satisfies Condition* (11)*, then it also satisfies Condition* (9)*.*

Both Conditions (9) and (11), as well as the expected payoff (Equation (7)), depend on the number of tasks n and self-predictor Δ_w. It is easy to show that the bounds on Δ_w in Conditions (9) and (11) are greater than or equal to 0. Since answer values are anticorrelated for a binary answer space, increase in belief for one answer value corresponds to the decrease in belief for another answer value. This means that, in a binary setting, Conditions (9) and (11) are satisfied regardless of n and Δ_w. In the next section, we show how the number of tasks influences the amount of positive correlation allowed between different answer values of a nonbinary answer space.

4.5. Limiting Cases with the Number of Tasks $n = 2$ and $n \to \infty$

We first examine the case when there is only one task in addition to a task t_w solved by a worker w. The expected payoff of a worker w with evaluation x for reporting y is, in that case, equal to

$$\alpha \cdot \left(\frac{Q_{p|w}(y|x)}{Q_p(y)} - 1 \right) \cdot (1 - (1 - Q_p(y))) = \alpha \cdot \left(Q_{p|w}(y|x) - Q_p(y) \right).$$

This means that, for $n = 2$, the RPTSC score is in expectation equivalent to the linear score

$$\tau(x_w, x_p) = \alpha \cdot (\mathbb{1}_{x_w = x_p} - R'_w(x_w)), \tag{12}$$

where $R'_w(x_w) = 2 \cdot (R_w(x_w) - \frac{\mathbb{1}_{x_w=x_p}}{2}) = \mathbb{1}_{x_w = x_q}$, that is, $R'_w(x_w)$ is constructed by sampling one report from the task not solved by worker w. The requirement for incentive compatibility of this score is

$$P_{p|w}(y|x) - P_p(y) < P_{p|w}(x|x) - P_p(x), \forall y \neq x.$$

That is, a worker's belief change from prior to posterior should be largest for the answer equal to the worker's evaluation. However, the condition for optimality (Condition (11)) imposes a restriction that different answer values are anticorrelated, that is:

$$P_{p|w}(y|x) - P_p(y) < 0, \forall y \neq x.$$

Although Condition (11) is only a sufficient condition of Theorem 4.7, it is actually tight for $n = 2$. Namely, if the condition did not hold, workers with evaluations x and y, and beliefs $P_{p|w}(y|x) - P_p(y) > 0$ and $P_{p|w}(x|y) - P_p(x) > 0$, would be better off reporting the same value (e.g., all report x or y) than reporting honestly. This comes from the fact that their expected payoff with such a collusive behavior would be $P_{p|w}(x|x) + P_{p|w}(y|x) - P_p(x) - P_p(y) \geq P_{p|w}(x|x) - P_p(x)$ and $P_{p|w}(x|y) + P_{p|w}(y|y) - P_p(x) - P_p(y) \geq P_{p|w}(y|y) - P_p(y)$, respectively, for workers with evaluations x and y.

With a larger number of tasks, we obtain that RPTSC mechanism has the same incentive compatibility requirements as the PTSC mechanism with reward function (1) - in that sense they are equivalent. In particular, the RPTSC requirements for incentive compatibility and optimality now coincide and are equal to the self-predicting condition with an unconstrained self-predictor $\Delta_w \in [0, 1]$. This means that for a larger number of tasks our mechanism allows a (bounded) positive correlation between two different answer values.

We see that a mechanism has to decide on an appropriate number of tasks to allow correlated answer values. To do this, it does not need knowledge about workers' beliefs, only estimates regarding the minimal value of priors $\min_{w,z} P_p(z)$ and the maximal value of self-predictors $\max_w \Delta_w$ among all workers w. The only restriction is that $\min_{w,z} P_p(z)$ is not overestimated and $\max_w \Delta_w$ is not underestimated. For example, one could incrementally take tasks into account, one by one, until workers' responses clearly indicate the minimal value of $\min_{w,z} P_p(z)$, determined by the frequency of the least frequent report, and the maximal value of $\max_w \Delta_w$, determined by the correlation among different reports.

Conditions (9) and (11) specify the upper bound on correlations among different answer values, described by self-predictor Δ_w, that implies incentive compatibility and optimality, given the number of tasks n. In the following table, we show how quickly the upper bound of Condition (11) approaches 1 as the number of tasks grows. Since, by Lemma 4.8, Condition (11) is stricter than Condition (9), the upper bound applies for both conditions. Clearly, for a reasonable number of tasks n, the bound allows significant correlation among different answers, even for the prior with values as small as 0.05.

n	$\min_z P_p(z) = 0.05$	$\min_z P_p(z) = 0.1$	$\min_z P_p(z) = 0.2$
10	$\Delta_w \leq 0.19$	$\Delta_w \leq 0.36$	$\Delta_w \leq 0.65$
30	$\Delta_w \leq 0.55$	$\Delta_w \leq 0.84$	$\Delta_w \leq 0.98$
60	$\Delta_w \leq 0.84$	$\Delta_w \leq 0.98$	$\Delta_w \leq 1$
100	$\Delta_w \leq 0.96$	$\Delta_w \leq 1$	$\Delta_w \leq 1$

We have seen that RPTSC reduces to a simple linear score when statistic R is calculated based on only one task in addition to the task being solved by a worker w. The form of the score (12) is similar to the Dasgupta&Ghosh mechanism introduced in Dasgupta and Ghosh [2013]. In fact, they are equivalent (see Section C in the Online Appendix), which means that the Dasgupta&Ghosh mechanism is a special case of RPTSC score obtained in the limit case when R is calculated from only two tasks. Moreover, the equivalence implies that the Dasgupta&Ghosh mechanism requires noncorrelated answer values for the honest-reporting strategy profile to result in the maximum profit.

4.6. Eliciting the Scaling Parameter of RPTSC

In the RPTSC, the expected reward for an honest answer, obtained by exerting effort, exceeds that for a heuristic answer by a positive margin. Using the scaling parameter α, this margin can be scaled so that it exceeds the cost of high effort. In this case, the expected reward for a heuristic strategy can be kept at 0, thus discouraging low effort workers from participating in the mechanism. We now show that it is possible to elicit a proper value of this parameter, for which workers who participate in solving tasks are incentivized to invest high effort.

Let us assume that strategies based on low effort induce some cost $c(e_0) > 0$. This can be interpreted, for example, as a cost of reporting. In order to achieve desirable properties, the RPTSC parameter α needs to be properly adjusted, so that the *a priori* expected payoff for investing high effort and honest reporting $\bar{\tau}_w(\alpha)$ satisfies $\bar{\tau}_w(\alpha) > c(e_1) - c(e_0)$. Although prescreening methods could be used to obtain proper scaling parameters, as discussed in Dasgupta and Ghosh [2013], we show how one can make use of potentially negative rewards in order to elicit the values of scaling parameters that would cover the cost of high effort.[4]

One way to elicit a proper value of α is by using auctioning, similar to Papakonstantinou et al. [2011]. We define the following two-step protocol:

(1) The mechanism asks workers to report the parameter α_w of the RPTSC mechanism that would recover their costs. After collecting their reports, it calculates $\hat{\alpha}$ such that π proportion of declared α_ws is greater than $\hat{\alpha}$. Then, every worker with α_w lower than $\hat{\alpha}$ proceeds to the next stage.
(2) Workers solve their tasks, report their answers, and are rewarded according to the RPTSC mechanism with $\alpha = \hat{\alpha}$.

To apply the protocol, we should have at least $\lceil \frac{2}{1-\pi} \rceil$ or more workers per task, where $\pi \in (0, 1)$. This guarantees that we have at least two workers per task when we execute RPTSC in the second stage of the mechanism, after removing a π proportion of the workers in the first phase.

[4]Negative rewards can, for example, be implemented by requesting that workers pay α for participation, then reward them with $\tau(x_w, x_p) + \alpha$, where τ is defined by RPTSC.

Because a worker would never announce a parameter α_i that leads to a negative profit, the two-step protocol is *ex-ante* individual rational and preserves the properties of the RPTSC.

PROPOSITION 4.9. *Consider workers with nontransferable utilities (payments). Suppose that* $c(e_0) > 0$ *and let* $\bar{\tau}_w(\alpha)$ *be a priori expected payoff of worker* w *in the honest-reporting equilibrium. Then, reporting in the first-stage* α_w *such that* $\bar{\tau}_w(\alpha_w) = c_w(e_1)$*, and using the honest-strategy profile in the second stage is an equilibrium of the two-step protocol.*

Note that, if workers are allowed to transfer their payments, then the auctioning model is not resilient to collusion. However, we can put a lower bound on the percentage of workers necessary to make a successful coalition by setting parameter π, which defines $\hat{\alpha}$, large enough. Indeed, in a realistic scenario, only a certain percentage of workers would form such coalitions. Thus, by having large π, we can avoid this type of collusion.

5. EXPERIMENTAL EVALUATION

In this section, we analyze the properties of the PTSC mechanisms in two crowdsourcing domains, community sensing and peer grading, focusing on two aspects: its manipulation resistance to collusive behavior and its effectiveness to elicit quality information. The analysis is done on a less robust version of the PTSC mechanism that requires a larger number of tasks, which usually holds in the proposed applications.

5.1. PTSC in Community Sensing

As an example of a community-sensing scenario, consider air-quality monitoring over an urban area. In order to use the PTSC mechanism, we need to identify the set of statistically independent tasks. Since air pollution is a localized phenomenon, we can define a set of tasks as measuring levels of air pollution at locations that are significantly away from each other. Assuming that measurements at these locations are conditionally independent given a global state, here denoted by Γ, we can apply the PTSC mechanisms in a similar fashion as described in Section 2. Global state Γ is modeled as a random variable that takes values in a finite discrete set $\{\gamma_1, \gamma_2, \ldots\}$. Note that the global state Γ does not describe how local variations influence a sensor's measurement at a certain location. For example, it might capture the fact that high humidity over an urban area increases the measured levels of air pollution, but it does not capture the fact that some places might have events such as traffic jams or fires.

We further assume that the self-predicting condition holds regardless of the global state Γ, that is:

$$\frac{P_{p|w}(y|x, \Gamma = \gamma)}{P_p(y|\Gamma = \gamma)} - 1 < \frac{P_{p|w}(x|x, \Gamma = \gamma)}{P_p(x|\Gamma = \gamma)} - 1, \forall \gamma \in \{\gamma_1, \gamma_2, \ldots\}, y \neq x.$$

In other words, the highest relative increase in a sensor's belief is obtained for the observed value. This condition is natural due to the fact that it describes the significance of sensors' measurements. With this in mind, we can reward a sensor using the PTSC mechanism with the following structure:

(1) For a sensor w that reports x_w, from a neighboring set of sensors (sensors located in the vicinity of sensor w), select a peer sensor p.
(2) Calculate the frequency of reports equal to x_w among reference sensors ρ ($|\rho| >> 1$), that include peer sensor p, but are not each other's neighbors:

$$R(x_w) = \frac{1}{|\rho|} \sum_{s \in \rho} \mathbb{1}_{x_w = x_s}.$$

(3) Reward sensor w with the PTSC mechanism of the form (1).

Since the reference sensors ρ are not each other's peers, their measurements are conditionally independent given Γ. Therefore, using the same reasoning as in the previous sections, we conclude that the expected payoff of sensor w, who observes x, for reporting y is approximately

$$\alpha \cdot \left(\sum_{\gamma \in \{\gamma_1, \gamma_2, \ldots\}} Pr(\Gamma = \gamma | x) \cdot \frac{P_{p|w}(y|x, \Gamma = \gamma)}{P_p(y|\Gamma = \gamma)} - 1 \right)$$

provided that the other sensors are honest. Similarly, when sensors adopt a strategy that is described by a distribution $Q_{p|p}$, the expected payoff of sensor w for reporting y is approximately

$$\alpha \cdot \left(\sum_{\gamma \in \{\gamma_1, \gamma_2, \ldots\}} Pr(\Gamma = \gamma | x) \cdot \frac{Q_{p|w}(y|x, \Gamma = \gamma)}{Q_p(y|\Gamma = \gamma)} - 1 \right).$$

Since $\frac{P_{p|w}(y|x,\Gamma=\gamma)}{P_p(y|\Gamma=\gamma)}$ satisfies the self-prediction for any global state γ, it is clear that the mechanism has the same properties as the PTSC mechanism presented in Section 2.

In order to apply the PTSC mechanism, one needs to find a peer measurement. In pollution sensing, we will rarely have two sensors in the same place; thus, we may use an artificial value constructed from the values reported by neighboring sensors. The simplest criteria for selecting neighbors would be to choose the m closest sensors. However, pollution can vary strongly due to differences in land use: a busy street will have much higher pollution than a forest next to it, even though the locations may be very close. This can be captured most accurately by models of pollution propagation, as we have done in earlier work [Faltings et al. 2014a]. In this experiment, we chose a simpler solution in which we just select one peer sensor in a location with similar characteristics. Finally, the center determines reference sensors ρ for each sensor. While this process can be a random selection with the constraint that reference sensors are not neighbors, in practice, one can consider ρ to be the set of all sensors, without affecting any incentive properties. Namely, it suffices that $R(x)$ converges to $P_p(x|\Gamma)$, which is for $|\rho| >> 1$ naturally satisfied in the considered setting.

We examine the characteristics of the presented PTSC using realistic data of nitrogen dioxide (NO_2) concentrations over the city of Strasbourg. The data consists of both real measurements collected by ASPA[5] and estimations of pollution from the physical model ADMS Urban V2.3 [Colvile et al. 2002]. In total, the dataset contains concentrations of NO_2 for each hour, expressed in parts per million (ppm), at 116 different locations over a period of 4 weeks.

Although the initial measurements take values in continuous domain, we discretize it using four levels of pollution defined as

—*low*: concentrations $0 - 20$ ppb;
—*medium*: concentrations $20 - 40$ ppb;
—*high*: concentrations $40 - 60$ ppb;
—*extrahigh*: concentrations $60 - \infty$ ppb.

Each hour, sensors report the measured level of pollution to the center and are rewarded with the PTSC mechanism (with constant $\alpha = 10$). As a criterion for neighbor selection, we consider distance and define neighbors of a certain sensor as 10 closest sensors. In peer selection, we effectively simulate the prior knowledge of the center by identifying

[5]www.atmo-alsace.net.

Table I. Average Payoffs (Stat. Sensors)

Strategy	Mean	Min	Max	Median	1st quartile	3rd quartile
honest	6.779	−003	59.969	3.658	2.703	7.806
collude	2.323	−0.146	21.769	1.045	0.7	2.805
colludeLow	0	0	0	0	0	0
random	0.022	−1.974	26.779	−1.076	−1.434	0.166
randomAll	0.071	−2.161	2.137	0.175	−0.438	0.609

for each location a neighboring location at which the true measurements are the most correlated to the true measurements at the considered location.[6] The sensor located in this neighboring location is considered as a peer. Empirical frequency $R(x)$ is calculated based on the reports of all sensors, except for the report of the sensor that is being scored.

To demonstrate the correctness of our results, we examine four different reporting strategies and evaluate their performance by analyzing the average scores of sensors. The four strategies are defined as follows:

—*honest*: All sensors are honest.
—*collude*: Sensors collude so that those who observe low or *medium* report low, while those who observe *high* or *extrahigh* report *high*.
—*colludeLow*: All sensors collude and report low.
—*random*: A sensor whose score is being calculated reports randomly with probabilities $Q_{w|w}(low) = \cdots = Q_{w|w}(extrahigh) = 0.25$; while other sensors are honest.
—*randomAll*: All sensors report randomly with probabilities $Q_{p|p}(low) = \cdots = Q_{p|p}(extrahigh) = 0.25$.

For each sensor, we run a separate process in which the sensors report according to one of these strategy profiles, and we calculate the average payoff of the considered sensor.

5.1.1. Stationary sensors. We first consider a scenario in which each sensor is stationary, that is, it occupies one location for the whole sensing period. The statistic of the average PTSC payoffs is shown in Table I. These payoffs can be further scaled in different ways, so that, for example, the incentives take positive values and cover the cost of sensing.

As expected, random reporting strategies lead to scores that are concentrated around 0, which is clearly seen from the median of *random* and *randomAll* strategies. Colluding on a single value results in a payoff equal to 0, which trivially follows from the structure of the score. The collusion strategy *collude* has a lower mean of the average payoffs than honest reporting. Moreover, a careful inspection of medians and quartiles shows that the collusive strategies are worse than honest reporting for the majority of sensors: the median, the 1st quartile, the 3rd quartile, and the maximum of average payoffs are greater for honest reporting than for the collusive strategies.

We tested the statistical significance of these results by applying the student's t-test on the average scores of sensors for each pair of the strategy profiles. The null hypothesis was that the average payoff of a sensor rewarded with different schemes follows the same distribution. As shown in Table II, the tests indicate that the distribution of average payoffs for truthful reporting is significantly different than the distributions of average payoffs for the other four strategy profiles, with p values less than 0.01.

[6]Note that we examine the correlations using the true data, not sensors' reports that are not necessarily truthful.

Table II. T-tests: p-Values for Different Pair Strategies (Stat. Sensors)

Strategy	*honest*	*collude*	*colludeLow*	*random*	*randomAll*
honest	-	$5 \cdot 10^{-7}$	$6 \cdot 10^{-14}$	$4 \cdot 10^{-13}$	$1 \cdot 10^{-13}$
collude	$5 \cdot 10^{-7}$	-	$3 \cdot 10^{-12}$	$1 \cdot 10^{-7}$	$2 \cdot 10^{-11}$
colludeLow	$6 \cdot 10^{-14}$	$3 \cdot 10^{-12}$	-	0.94	0.301
random	$4 \cdot 10^{-13}$	$1 \cdot 10^{-7}$	0.94	-	0.873
randomAll	$1 \cdot 10^{-13}$	$2 \cdot 10^{-11}$	0.301	0.873	-

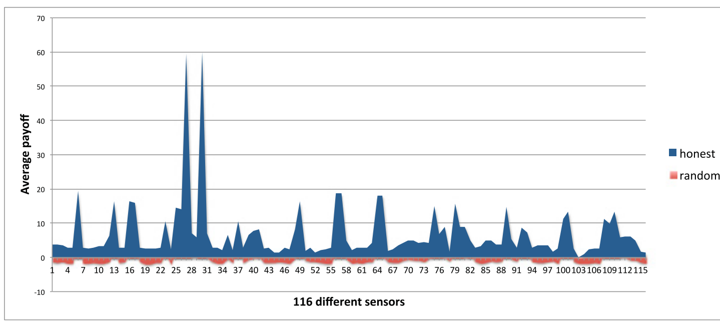

Fig. 2. Average payoffs of *honest* and *random* strategies for each sensor (stat. sensors), arranged in no particular order along the x-axis.

Table III. Average Payoffs (Mobile Sensors)

Strategy	Mean	Min	Max	Median	1st quartile	3rd quartile
honest	6.779	4.064	12.941	6.456	5.665	7.664
collude	2.323	1.052	5.141	2.027	1.755	2.707
colludeLow	0	0	0	0	0	0
random	−0.008	−1.781	3.714	−0.294	−0.702	0.389
randomAll	0.03	−1.446	1.792	−0.109	−0.38	0.49

We obtain a similar result for the strategy *collude*. For the other three strategy profiles, one cannot reject the null hypothesis with the significance level $\alpha = 0.05$. This is not surprising considering that, for these three strategies, the average payoffs are concentrated around 0.

The described scenario involves stationary sensors, so the sensors were solving approximately the same task over a longer period. This deviates from our assumption that tasks are randomly distributed to workers, thus a sensor that reports randomly might obtain a relatively high average payoff over a longer sensing period when the histogram of its reports is more correlated to the reports of its peer than R is. Although a sensor reports randomly, its reports carry some information about its peer with regard to R; thus, it is not surprising that such a sensor might obtain positive rewards. Note, however, that honest reporting leads to significantly higher payoffs, as shown in Figure 2.

5.1.2. Mobile sensors. In community sensing, sensors are often mobile, in which case the assumption about random task assignment is approximately fulfilled. Therefore, we further investigate what happens when sensors' locations are randomly selected at each iteration. As shown by the statistic of the average PTSC payoffs in Table III and p values in Table IV, the qualitative results are similar to the stationary case,

Table IV. T-tests: p Values for Different Pairs of Strategies (Mobile Sensors)

Strategy	*honest*	*collude*	*colludeLow*	*random*	*randomAll*
honest	-	$1 \cdot 10^{-60}$	$7 \cdot 10^{-74}$	$2 \cdot 10^{-90}$	$6 \cdot 10^{-84}$
collude	$1 \cdot 10^{-60}$	-	$5 \cdot 10^{-57}$	$3 \cdot 10^{-48}$	$6 \cdot 10^{-61}$
colludeLow	$7 \cdot 10^{-74}$	$5 \cdot 10^{-57}$	-	0.933	0.637
random	$2 \cdot 10^{-90}$	$3 \cdot 10^{-48}$	0.933	-	0.739
randomAll	$6 \cdot 10^{-84}$	$6 \cdot 10^{-61}$	0.637	0.739	-

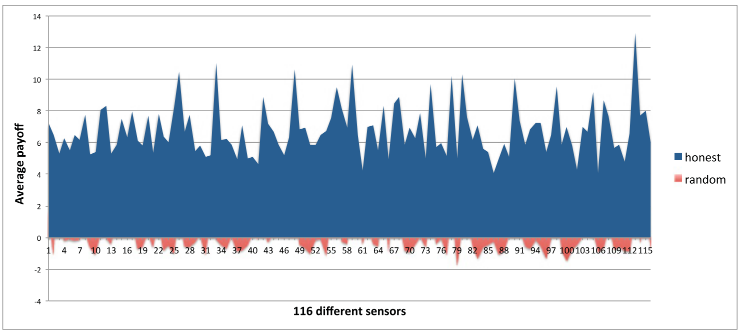

Fig. 3. Average payoffs of *honest* and *random* strategies for each sensor (mobile sensors), arranged in no particular order along the x-axis.

with the main difference being the variance of the average payoffs.[7] Furthermore, the honest-reporting strategy results in significantly higher payoffs than random reporting (Figure 3), as it is the case for stationary sensors.

5.2. PTSC in Peer Grading

In order to test the impact that PTSC has on the quality of grades, we designed a peer-grading experiment within an "Artificial Intelligence" course at EPFL. In particular, as a part of the evaluation process, the course contained three quizzes, each consisting of two parts: in one part, students were asked to add a missing code; in the other, they were asked to find mistakes in a given code. The three quizzes took place at different time periods during the semester, assessing the knowledge about different topics of the course. Each problem in the quizzes had a correct solution. These solutions were used to assign points to the students, which were a part of the final grade. The official corrections of the quizzes were done by the teaching assistants of the course. Before the official points were announced, the students were asked to correct the solutions of their colleagues based on the correct solutions.

A criterion to determine the quality of a solution for a part of a quiz in which students were supposed to add a missing code was described by three to four different cases that defined potential mistakes or shortcomings of a student's solution. These cases were designed so that each covers a combination of possibilities that can occur in students' solutions, keeping in mind that the combinations are mutually exclusive between the cases. Naturally, a peer grader was selecting only one of these cases, and reporting only one value in total for the whole part. For the other part of the quiz, in which students were supposed to find mistakes in a given code and correct them, a grading criterion

[7] The mean payoffs for *honest* and *collude* are identical for stationary (Table I) and mobile sensors (Table III) simply because the means are calculated on the same set of values (the only difference is that, in the mobile scenario, these values are permuted). For the random reporting strategies, these means are different because we ran two different experiments over two different (randomly generated) sets of reports.

was much easier to define. For each mistake in a given code, a student could either: not find the mistake; find a mistake, but not correct it; or find a mistake and correct it. Therefore, a peer grader was presented with these three possibilities. Note, however, that a peer grader made such reports for all mistakes that were in a given code (four to five), effectively reporting several values. Each reported value was separately treated in a peer-rewarding mechanism.

To incentivize participation, we rewarded peer graders with bonus points (additional points that could improve their grades), that were obtained using one of the three different reward schemes: a *constant* reward, a *peer consistency* [Huang and Fu 2013b], and PTSC. For the constant-reward regime, a peer grader who participated in the peer grading obtained the maximum number of bonus points $MaxTotalReward$. For the peer consistency, reward for reporting an answer was equal to $\frac{MaxTotalReward}{NumTasks}$ if a chosen peer reports the same answer, and was 0 otherwise. $NumTasks$ denotes the number of subparts to grade, which was equal to the number of reports that a peer grader made. The PTSC mechanism was also applied for each report separately. Furthermore, the scaling parameter of PTSC was equal to $\alpha = \frac{1}{2} \cdot \frac{MaxTotalReward}{NumTasks}$, and the scores were shifted (increased) by α so that the bonus points turn out positive. If a total number of the PTSC points exceeded $MaxTotalReward$, it was set to $MaxTotalReward$. Finally, statistic $R(x)$ in PTSC was designed for each subpart of a quiz separately; it was defined as an empirical frequency of grades equal to x among all reports that are rewarded with PTSC for that subpart of the quiz.

To test the quality of the reward schemes, we split the students into three groups of approximately the same number of students. Since participation in the peer grading experiment was not obligatory, the sizes of these groups varied. Each group was rewarded using all three reward schemes, but different mechanisms were applied for different quizzes in a round robin fashion. That is: if PTSC was used to assign rewards to a group for peer grading the first quiz, the same group was rewarded with the constant reward for peer grading the second quiz; if the peer consistency was used to assign rewards to a group for peer grading the first quiz, the same group was rewarded by PTSC for peer grading the second quiz, and so on.

In order to do a peer grading for a quiz, students needed to go through a tutorial that explained the peer grading task and a reward scheme that was used to assign bonus points; these were separately explained in two different sections. The tutorial also contained two examples, one for the task explanation and one for the mechanism explanation. Each example contained a simple test question for improving students' understanding. Different schemes had a different example question, showing the most basic features of the mechanisms. For the constant reward, students were asked to answer how many points they would obtain upon fulfilling the peer grading task, with three possible answers: *MaxTotalReward per task*, *MaxTotalReward in total*, or *it depends on how other raters grade*. For the peer consistency, the question asked to pick the correct claim, provided that the peer reported *correct*. The claims were *for reporting correct the reward is* 0, *for reporting incorrect the reward is* $\frac{MaxTotalReward}{NumTasks}$, or *for reporting correct the reward is* $\frac{MaxTotalReward}{NumTasks}$. Finally, for PTSC, the question asked what the reward was for reporting *correct* provided that everybody else reported *correct*. The options were $5 \cdot \frac{MaxTotalReward}{2 \cdot NumTasks}$, $3 \cdot \frac{MaxTotalReward}{2 \cdot NumTasks}$, or $1 \cdot \frac{MaxTotalReward}{2 \cdot NumTasks}$.[8] The options for each question were presented in a different order for different groups.

We measured the quality of raw data (nonaggregated responses from students) with respect to the corrections made by the teaching assistants. For each student, we calculated the number of correct reports. Then, for each mechanism, we determined the

[8]We used numerical values in all three of the test questions.

Table V. Average Error Rate for Different Mechanisms

Mechanism	Number of students	Error rate (%)
PTSC	16	6.88
peer consistency	16	10.48
constant	14	11.98

Table VI. T-tests: p Values for Different Mechanisms

Mechanism	PTSC	Peer consistency	Constant
PTSC	-	0.0255	0.0497
peer consistency	0.0255	-	0.5566
constant	0.0497	0.5566	-

average error rate, that is, the percentage of incorrect grades. To measure the statistical significance, we performed two-tailed student's t-test, with the significance level of 0.05. The null hypothesis was that students' error rates for two groups rewarded by different mechanisms follow the same distribution.

For the first two quizzes, each peer grader graded 4 partial solutions of one's colleagues: 2 solutions to the first part of the quiz and 2 solutions to the second part of the quiz. Since our analysis did not reveal any statistical significance of the accuracy of the raw data across different schemes, we increased the number of solutions to grade for the third peer-grading task. That is, for the third quiz, each peer grader graded 10 partial solutions of one's colleagues: 5 solutions to the first part of the quiz and 5 solutions to the second part of the quiz.

The results of the third quiz are shown in Table V; for each group, they contain the number of students and the average error rate. As we can see, PTSC outperforms the baseline algorithms by 3% to 5%. Furthermore, t-tests (in Table VI) show that there is a statistically significant difference between the error rates for the PTSC mechanism and the error rates for the constant reward or the peer consistency, with p values equal to 0.0497 and 0.0255, respectively.

6. CONCLUSIONS

Ensuring the accuracy of answers provided by workers is a major challenge for using crowds as part of intelligent systems. Instead of paying fixed rewards, it is desirable to use a payment scheme for which only workers that actually solve the tasks can expect to receive a reward. Such a scheme is important for two reasons:

—to improve the accuracy of answers, thus complement filtering mechanisms such as gold tasks and reputation systems, and
—to make worker self-selection help the mechanism by discouraging workers that do not contribute useful results.

We show a new incentive scheme, called the Peer Truth Serum for Crowdsourcing (PTSC), in which strategies that result in positive expected payments require workers to solve the tasks and truthfully report the answers. PTSC is the first mechanism that applies to problems with more than 2 answers and heterogeneous worker populations. We show under which conditions the mechanism has strong incentive properties and how to elicit the scaling parameters to achieve those properties. Due to its simplicity and robustness, our payment scheme is applicable to a wide variety of crowdsourcing settings, such as community sensing or peer grading in massive open online courses.

When the number of tasks is equal to 2, the robust version of PTSC (RPTSC) is equivalent to the Dasgupta&Ghosh mechanism [Dasgupta and Ghosh 2013]. However, when there is a significant number of tasks in a batch, the nonlinear structure of

RPTSC allows correlations among different answer values, within the bounds of the self-predicting condition. From a practical point of view, an interesting direction for future work is to investigate an adaptive mechanism that would automatically determine the number of tasks needed to have sufficiently loose conditions for incentive compatibility and optimality.

Another direction is to investigate the structure of the payments for large answer spaces. Namely, the RPTSC has strong incentive properties in terms of a worker's *expected* payoff. For smaller answer spaces, the concentration inequalities, such as Hoeffding's inequality, imply that the overall payoff will very likely be close to the expected value whenever a worker solves a reasonable number of tasks. For larger answer spaces, one also needs to lower the variance of the RPTSC score; this can potentially be achieved by investigating correlations among different possible answers, similar to how Radanovic and Faltings [2014] designed a peer-rewarding mechanism for elicitation of continuous signals.

ACKNOWLEDGMENTS

We thank Jason Jingshi Li for providing a community-sensing testbed and the anonymous reviewers for useful comments and feedback.

REFERENCES

Luis Von Ahn and Laura Dabbish. 2004. Labeling images with a computer game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

R. N. Colvile, N. K. Woodfield, D. J. Carruthers, B. E. A. Fisher, A. Rickard, S. Neville, and A. Hughes. 2002. Uncertainty in dispersion modeling and urban air quality mapping. *Environmental Science and Policy* 5, 207–220.

Anirban Dasgupta and Arpita Ghosh. 2013. Crowdsourced judgement elicitation with endogenous proficiency. In *Proceedings of the 22nd ACM International World Wide Web Conference (WWW13)*.

Boi Faltings, Jason J. Li, and Radu Jurca. 2014a. Incentive mechanisms for community sensing. *IEEE Transactions on Computers* 63, 115–128.

Boi Faltings, Pearl Pu, Bao Duy Tran, and Radu Jurca. 2014b. Incentives to counter bias in human computation. In *Proceedings of the 2nd AAAI Conference on Human Computation and Crowdsourcing (HCOMP'14)*.

Xi Alice Gao, Andrew Mao, and Yiling Chen. 2013. Trick or treat: Putting peer prediction to the test. In *Workshop on Crowdsourcing and Online Behavioral Experiments, in Conjunction with ACM EC13*.

Florent Garcin and Boi Faltings. 2014. Swissnoise: Online polls with game-theoretic incentives. *Proceedings of the 26th Conference on Innovative Applications of AI*, 2972–2977.

Sharad Goel, Daniel M. Reeves, and David M. Pennock. 2009. Collective revelation: A mechanism for self-verified, weighted, and truthful predictions. In *Proceedings of the 10th ACM Conference on Electronic Commerce (EC'09)*.

Christopher Harris. 2011. You're hired! An examination of crowdsourcing incentive models in human resource tasks. In *Proceedings of the Workshop on Crowdsourcing for Search and Data Mining (CSDM) at the 4th ACM International Conference on Web Search and Data Mining (WSDM'11)*.

Shih-Wen Huang and Wai-Tat Fu. 2013a. Don't hide in the crowd!: Increasing social transparency between peer workers improves crowdsourcing outcomes. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

Shih-Wen Huang and Wai-Tat Fu. 2013b. Enhancing reliability using peer consistency evaluation in human computation. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*.

Radu Jurca and Boi Faltings. 2009. Mechanisms for making crowds truthful. *Journal of Artificial Intelligence Research (JAIR)* 34, 209–253.

Radu Jurca and Boi Faltings. 2011. Incentives for answering hypothetical questions. In *Workshop on Social Computing and User Generated Content (EC'11)*.

Ece Kamar and Eric Horvitz. 2012. Incentives for truthful reporting in crowdsourcing. In *Proceedings of the 2012 International Conference on Autonomous Agents and Multiagent Systems (AAMAS'12)*.

Nicolas Lambert and Yoav Shoham. 2008. Truthful surveys. In *Proceedings of the 3rd International Workshop on Internet and Network Economics (WINE'08)*.

Nolan Miller, Paul Resnick, and Richard Zeckhauser. 2005. Eliciting informative feedback: The peer-prediction method. *Management Science* 51, 1359–1373.

Athanasios Papakonstantinou, Alex Rogers, Enrico Gerding, and Nicholas Jennings. 2011. Mechanism design for the truthful elicitation of costly probabilistic estimates in distributed information systems. *Artificial Intelligence* 175, 648–672.

Drazen Prelec. 2004. A bayesian truth serum for subjective data. *Science* 34, 5695 (2004), 462–466.

Drazen Prelec and Sebastian Seung. 2006. An algorithm that finds truth even if most people are wrong. Working paper.

Goran Radanovic and Boi Faltings. 2013. A robust Bayesian truth serum for non-binary signals. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence (AAAI'13)*.

Goran Radanovic and Boi Faltings. 2014. Incentives for truthful information elicitation of continuous signals. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI'14)*.

Goran Radanovic and Boi Faltings. 2015a. Incentive schemes for participatory sensing. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems (AAMAS'15)*.

Goran Radanovic and Boi Faltings. 2015b. Incentives for subjective evaluations with private beliefs. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15)*.

Aaron Shaw, Daniel L. Chen, and John Horton. 2011. Designing incentives for inexpert human raters. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'11)*.

Adish Singla and Andreas Krause. 2013. Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In *Proceedings of the 22nd International Conference on World Wide Web*.

Hossein Azari Soufiani, David C. Parkes, and Lirong Xia. 2012. Random utility theory for social choice. In *Proceedings of Neural Information Processing Systems*.

Bo Waggoner and Yiling Chen. 2013. Information elicitation sans verification. In *Information Elicitation Sans Verification. In Proceedings of the 3rd Workshop on Social Computing and User Generated Content (SC'13)*.

Bo Waggoner and Yiling Chen. 2014. Output agreement mechanisms and common knowledge. In *Proceedings of the 2nd AAAI Conference on Human Computation and Crowdsourcing (HCOMP'14)*.

Jens Witkowski, Yoram Bachrach, Peter Key, and David C. Parkes. 2013. Dwelling on the negative: Incentivizing effort in peer prediction. In *Proceedings of the 1st AAAI Conference on Human Computation and Crowdsourcing*.

Jens Witkowski and David C. Parkes. 2012a. Peer prediction without a common prior. In *Proceedings of the 13th ACM Conference on Electronic Commerce (EC'12)*. 964–981.

Jens Witkowski and David C. Parkes. 2012b. A robust Bayesian truth serum for small populations. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI'12)*.

Jens Witkowski and David C. Parkes. 2013. Learning the prior in minimal peer prediction. In *Proceedings of the 3rd Workshop on Social Computing and User Generated Content (SC'13)*.

Peter Zhang and Yiling Chen. 2014. Elicitability and knowledge-free elicitation with peer prediction. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multiagent Systems (AAMAS'14)*.

Received January 2015; revised August 2015; accepted November 2015