Learning to Scale Payments in Crowdsourcing with PropeRBoost

Goran Radanovic
Ecole Polytechnique Federale de Lausanne
(EPFL)
Artificial Intelligence Laboratory
CH-1015 Lausanne, Switzerland
goran.radanovic@epfl.ch

Boi Faltings
Ecole Polytechnique Federale de Lausanne
(EPFL)
Artificial Intelligence Laboratory
CH-1015 Lausanne, Switzerland
boi.faltings@epfl.ch

Abstract

Motivating workers to provide significant effort has been recognized as an important issue in crowdsourcing. It is important not only to compensate worker effort, but also to discourage low-quality workers from participating. Several proper incentive schemes have been proposed for this purpose; they are either based on gold tasks or on peer consistency in individual tasks. As the rewards cannot become negative, these schemes have difficulty in achieving zero expected reward for random answers.

We describe a novel boosting scheme, ProperRBoost, that improves the efficiency of existing incentive schemes by making a better separation between incentives for high and low quality work, and effectively discourages random answers by assigning them near minimal average rewards. We show the actual performance of the boosting scheme through simulations of various worker strategies.

Introduction

One of the main issues in crowdwork is the existence of *spam* workers, or shortly *spammers*, who provide inaccurate or random data. Since providing accurate data requires effort, it comes as no surprise that workers are inclined to deviate by investing as little effort as possible, and, hence, reporting inaccurate information. One of the basic approaches in incentivizing workers to invest high effort is to provide them with (monetary) rewards that would compensate their cost of solving tasks.

Recently, many mechanisms have been proposed for assigning rewards to workers based on their performance in solving micro-tasks. These mechanisms are either based on gold tasks (test tasks whose correct answers are known) (Oleson et al. 2011; Harris 2011; Shah, Zhou, and Peres 2015; Shah and Zhou 2015) or are peer-consistency-based incentives (Dasgupta and Ghosh 2013; Radanovic, Faltings, and Jurca 2016; Shnayder et al. 2016), suitable for elicitation of subjective information (Miller, Resnick, and Zeckhauser 2005; Prelec 2004; Witkowski and Parkes 2012b; 2012a; Radanovic and Faltings 2013). The basic principle of any strictly *proper (incentive-compatible)* payment mechanism is to place higher rewards to more accurate answers. Since the accuracy of an answer reflects the effort invested in obtaining it, a proper payment mechanism clearly assigns higher rewards to *good workers* who put in high effort and give truthful answers. A spammer strategy will still occasionally hit the correct answer and result in a high reward. Either these occurrences are compensated by charging a worker to participate, or the property of making a spammer's expected reward equal to zero cannot be achieved.

A traditional approach of discouraging spammers to participate is by transforming the incentives using an affine transformation so that in expectation they are equal to or less than 0 for spam workers. While such an approach might have a positive impact in terms of a mechanism's profit (Witkowski et al. 2013), negative payments are not easy to implement in practice, and can even deter good workers from participating. To minimize the expected payments to spammers when only positive payments are allowed, one can use the multiplicative incentive mechanism of (Shah and Zhou 2015), which implements the *double-or-nothing* principle using gold tasks. In particular, for every correct answer in the gold standard questions, the reward doubles, while if any of the gold standard question are answered incorrectly, the reward is 0. Furthermore, a worker can choose to skip questions.

The mentioned performance-based mechanisms, however, do not optimize the difference between the expected payment for good work and the expected payment for spamming, which is important because low quality work typically costs less than accurate reporting in terms of effort and time. For many of these mechanisms, this difference can be quite small, especially when the elicitation setting contains significant amount of noise. For example, in crowd-sensing, measurements are often noisy, or in the peer-consistency-based mechanisms, peer reports might come from both good workers and spammers.

To address this issue, we examine a scenario in which workers interact with a mechanism over a longer period of time. Our goal is to design a mechanism that would output scaling factors in interval $[0, 1]$ that can be used as multiplicative factors of positive payments. A mechanism should provide scales such that the average payment of a good worker converges to the average payment with the scale equal to 1, and the average payment of a spammer converges to 0. We note that this objective differs from a traditional approach of maximizing surplus or a requester's profit, as often done in similar principal-agent models in economics literature (e.g., see (MacLeod and Malcomson 1989; Levin 2003; Ho, Slivkins, and Vaughan 2014)). However, we believe that the differentiation between payments for good work and spamming plays an important role in both self-selection of workers and encouraging workers to provide accurate information.

Contribution: Using techniques from the online learning and reputation systems literature (e.g., (Nisan et al. 2007; Cesa-Bianchi and Lugosi 2006; Resnick and Sami 2007; Radanovic and Faltings 2016)), we develop a novel way of boosting payments that can improve existing incentive mechanisms. We propose a two stage reputation protocol, that we call PropeRBoost, in which: 1) the reputation of a worker defines how her payment is scaled; 2) the reputation of a worker increases or decreases depending on the quality of the work that the worker provides. Therefore, the reward for achieving a good performance is reflected through a reward for a later task. This allows us to penalize bad behavior by deducting from future payments, and thus boost the payment gap between good and bad worker behavior. To evaluate the performance of the mechanism, we define a utility measure that captures how well scaling factors reward accurate work and suppress rewards for low quality work.

In the paper, we focus on reports that have a binary information structure to show that the average PropeRBoost payments are near maximum for good workers and near minimum for spammers. We provide simulation results, which reveal that our design leads to a greater separation between payments for good work and spamming than the one obtained with a standard incentive design.

Preliminaries

We are interested in a crowdsourcing scenario where workers repeatedly solve micro-tasks of the same difficulty, such as image labeling, product reviewing or sensing. Our goal is to design a mechanism that would output scaled payments so that low quality workers obtain average rewards close to zero, while the rewards of high quality workers are scaled to maximum value. The scale of a worker is denoted by σ and it takes values in $[0, 1]$. We formalize the setting as follows.

In our setting, a mechanism periodically posts micro-tasks that are assigned to a group workers W. Notice that the question of how tasks are distributed among the workers in W is application dependent. For example, at a certain time step, there might be 100 tasks assigned to 50 workers, each worker solving 12 tasks. We will, thus, consider a mechanism from the perspective of 1 particular worker and examine how her reward is scaled in a long run, depending on the reporting strategy adopted by the worker. We denote this worker by w and associate to her time t, which measures how many times the worker participated in the mechanism at a certain point of time. We denote by T the expected number of times that the worker will interact with the mechanism.

A worker w is considered to be a *long-lived agent* that evaluates each of the tasks given to her with x or y. While the binary information model has its limitations, it is often used in recent literature on incentives (e.g., see (Dasgupta and Ghosh 2013; Witkowski and Parkes 2012b; 2012a)), and can often be applied to non-binary information sets, as we demonstrate in our simulations. To describe how worker w evaluates her tasks, we use a simple probabilistic model that assumes the existence of the ground truth. This is a standard approach when dealing with objective information (e.g., (Dasgupta and Ghosh 2013)) and a common approach when dealing with subjective information (e.g. (Soufani, Parkes, and Xia 2012)).

In particular, we assume that for each task, there is a correct answer $\theta \in \{x, y\}$, and a worker endorses a noisy evaluation of θ that represents her answer to the task, which we denoted by X. As in (Dasgupta and Ghosh 2013), we define the *proficiency* of a worker as the probability of the worker being correct, i.e., $\Pr(X = \theta)$ — throughout the paper we denote it by p. It is reasonable to assume that a worker is at least as good as a random guess, i.e., $p \geq \frac{1}{2}$. Nevertheless, worker w can be dishonest, and this situation corresponds to the one in which worker w is honest, but has proficiency level $1 - p$. Therefore, we allow p to take values in the whole interval $[0, 1]$, which enables us to model both worker's strategy and her quality via p.

Notice that a worker's proficiency is closely tied to her performance in solving tasks. Therefore, in a long run, a mechanism should output scaling $\sigma_w \approx 0$ if worker w has an average performance lower than a proficiency threshold p_l. We also define a threshold p_u that defines a maximal level of worker w's proficiency for which σ_w might not be in expectation approximately equal to 1 in a long run. Thresholds p_l and p_u are parameters set by a mechanism and are assumed to relate as $p_u > p_l \geq \frac{1}{2}$. Intuitively, p_l defines the acceptable level of proficiency, while p_u defines the good level of proficiency. For example, we can set that $p_l = 0.7$, while $p_u = 0.8$. In general, p_u can be arbitrarily close to p_l, but the more separate they are, the easier it is to distinguish proficient workers from those that provide low quality answers.

Furthermore, we consider an estimator $\mathcal{F}$ that produces estimates of the correct answers for each task that we use to evaluate worker w. We assume that the estimator performs better that a random guess, i.e., independently for each task we have that $\Pr(\hat{\theta}_{\mathcal{F}} = \theta) > \frac{1}{2}$, where $\hat{\theta}_{\mathcal{F}}$ is an estimate of $\mathcal{F}$. Accuracy $\Pr(\hat{\theta}_{\mathcal{F}} = \theta)$ and the prior bias of the correct answers $(2 \cdot \Pr(\theta = x) - 1)^2$ are assumed to be known. We denote $\Pr(\hat{\theta}_{\mathcal{F}} = \theta)$ by q and $(2 \cdot \Pr(\theta = x) - 1)^2$ by γ.

As we demonstrate in Section *Simulations*, having an estimator with $q > \frac{1}{2}$ is a reasonable assumption, even if we do not have gold standard tasks. For example, the condition will typically hold for many implementations of $\mathcal{F}$ (e.g., (Raykar et al. 2010; Karger, Oh, and Shah 2011; Liu, Peng, and Ihler 2012; Karger, Oh, and Shah 2013)) if workers are not malicious, although some of them might be spammers, as in the *spammer-hammer* model described by (Karger, Oh, and Shah 2011).

Performance Metric

In order to provide a good scaling factors, a mechanism has to make a proper evaluation of a worker's performance. We measure a worker's performance on a task by how much more accurate her report $X = z$ is than it is expected by a prior distribution $\Pr(\theta)$:

$$\pi_w = \mathbb{1}_{\theta=z} - \Pr(\theta = z)$$

where $\mathbb{1}$ is an indicator variable equal to 1 when $\theta = z$ and 0 otherwise. Since the mechanism does not have knowledge of θ, but only its estimate $\hat{\theta}_{\mathcal{F}}$, we further define the estimated accuracy for report $X = z$ as:

$$\hat{\pi}_w = \mathbb{1}_{\hat{\theta}_{\mathcal{F}}=z} - \hat{\pi}_{\Pr(\hat{\theta}_{\mathcal{F}}=z)} \tag{1}$$

where $\hat{\pi}_{\Pr(\hat{\theta}_{\mathcal{F}}=z)}$ is either prior $\Pr(\hat{\theta}_{\mathcal{F}} = z)$ or a random variable whose expected value is equal to $\Pr(\hat{\theta}_{\mathcal{F}} = z)$. The letter is often available in a typical crowdsourcing scenario, as shown in Section *Simulations*. The following holds:

Proposition 1. *For worker w with proficiency p, the expected value of $\hat{\pi}_w$ for report X is equal to:*

$$\mathbb{E}(\hat{\pi}_w) = 2 \cdot (1-\gamma) \cdot (q - \frac{1}{2}) \cdot (p - \frac{1}{2})$$

Proof. Let $X = z \in \{x, y\}$. The expected value of $\mathbb{1}_{\hat{\theta}_{\mathcal{F}}=z}$ is equal to:

$$\Pr(\hat{\theta}_{\mathcal{F}} = \theta) \cdot \Pr(\theta = X) + \Pr(\hat{\theta}_{\mathcal{F}} \neq \theta) \cdot \Pr(\theta \neq X) =$$
$$= q \cdot p + (1-q) \cdot (1-p) = 2 \cdot (q - \frac{1}{2}) \cdot (p - \frac{1}{2}) + \frac{1}{2}$$

Prior $\Pr(\hat{\theta}_{\mathcal{F}} = z')$ for $z' \in \{x, y\}$ is equal to:

$$\begin{aligned} &\Pr(\hat{\theta}_{\mathcal{F}} = \theta) \cdot \Pr(\theta = z') \\ &\quad + (1 - \Pr(\hat{\theta}_{\mathcal{F}} = \theta)) \cdot (1 - \Pr(\theta = z')) \\ &\quad = 2 \cdot (q - \frac{1}{2}) \cdot (\Pr(\theta = z') - \frac{1}{2}) + \frac{1}{2} \end{aligned}$$

and, similarly, we obtain $\Pr(X = z')$:

$$\Pr(X = z') = 2 \cdot (p - \frac{1}{2}) \cdot (\Pr(\theta = z') - \frac{1}{2}) + \frac{1}{2}$$

The expectation of $\Pr(\hat{\theta}_{\mathcal{F}} = z)$ over the possible values of X is:

$$\begin{aligned} &\Pr(X = x) \cdot \Pr(\hat{\theta}_{\mathcal{F}} = x) + \Pr(X = y) \cdot \Pr(\hat{\theta}_{\mathcal{F}} = y) \\ &= 4 \cdot (p - \frac{1}{2}) \cdot (q - \frac{1}{2}) \cdot [(\Pr(\theta = x) - \frac{1}{2})^2 \\ &+ (\Pr(\theta = y) - \frac{1}{2})^2] + \frac{1}{2} \\ &= 2 \cdot (p - \frac{1}{2}) \cdot (q - \frac{1}{2}) \cdot (2 \cdot \Pr(\theta = x) - 1)^2 + \frac{1}{2} \\ &= 2 \cdot \gamma \cdot (p - \frac{1}{2}) \cdot (q - \frac{1}{2}) + \frac{1}{2} \end{aligned}$$

By combining all the expressions above with (1) and noting that $\mathbb{E}(\hat{\pi}_{\Pr(\hat{\theta}_{\mathcal{F}}=z)}) = \mathbb{E}(\Pr(\hat{\theta}_{\mathcal{F}} = z))$, we obtain:

$$\begin{aligned} \mathbb{E}(\hat{\pi}_w) &= \mathbb{E}(\mathbb{1}_{X=\hat{\theta}_{\mathcal{F}}}) - \mathbb{E}(\hat{\pi}_{\Pr(\hat{\theta}_{\mathcal{F}}=X)}) \\ &= 2 \cdot (1-\gamma) \cdot (p - \frac{1}{2}) \cdot (q - \frac{1}{2}) \end{aligned}$$

□

Since parameter q is greater than $\frac{1}{2}$, from Proposition 1 follows that the expected value of performance metric $\hat{\pi}_w$ is maximized when worker w is honest and as accurate as possible.

Score Function

Notice that the performance metric $\hat{\pi}_w$ is linear in proficiency p of worker w. This enables us to assign a quality score to a worker w that directly tells us how much more proficient worker w is than the lowest acceptable proficiency level p_l. In particular, we define the score of a worker w for a particular task as:

$$score_w = (1 - \alpha) \cdot \hat{\pi}_w - \alpha \tag{2}$$

where α is a predefined parameter. The direct consequence of Proposition 1 is:

Corollary 1. *Let $\alpha = \frac{2 \cdot (1-\gamma) \cdot (p_l - \frac{1}{2}) \cdot (q - \frac{1}{2})}{1 + 2 \cdot (1-\gamma) \cdot (p_l - \frac{1}{2}) \cdot (q - \frac{1}{2})}$. Then, for worker w with proficiency p, the score defined by expression* (2) *is in expectation equal to:*

$$\mathbb{E}(score_w) = A \cdot (p - p_l)$$

where:

$$A = \frac{2 \cdot (1-\gamma) \cdot (q - \frac{1}{2})}{1 + 2 \cdot (1-\gamma) \cdot (p_l - \frac{1}{2}) \cdot (q - \frac{1}{2})}$$

Utility Function

It remains to define an objective (utility) function of a scaling mechanism that provides scaling factors σ_w. As noted in the previous sections, we investigate how to maximally separate average payments for low and good quality work. Therefore, we define the gain g of a mechanism as:

$$g(\sigma_w, p) = \sigma_w \cdot (p - p_l) \tag{3}$$

In other words, the greater the scaling is for a worker with good performance ($p > p_l$), the better the performance of the mechanisms is. Likewise, the lower the scaling is for a worker who provides low quality work ($p < p_l$), the better the performance of the mechanism is. The gain is positive for good quality work and negative for low quality work. Thus, it describes the performance of the mechanisms relative to the rewarding mechanism that assigns 0 rewards to everyone.

PropeRBoost

We now turn to our main contribution: a reputation system that limits the effectiveness of spam workers whose reporting strategies are based on low effort. We call the mechanism *PropeRBoost* and it has the structure defined by Algorithm 1. As noted in the preliminaries, we focus on a particular worker w that interacts with the mechanism over a long period.

Worker w initially has reputation score equal to ρ_0, where ρ_0 is a small positive number. Then, over a long time period T, the worker solves different tasks and reports her answers to the mechanism. Once worker w solves a task at time t, the algorithm estimates the correct answer to that task, $\hat{\theta}_{\mathcal{F}}$, and calculates the performance $\hat{\pi}_w$ using expression (1). This performance further defines $score_w$, which is used to update the reputation of worker w. Notice that worker w is allowed to solve multiple tasks at one time period. In that case, we use the average of all the scores $score_w$ in the reputation updating procedure. Due to the linearity of the expectation, we know that the statement of Corollary 1 holds for the average as well. Furthermore, if multiple tasks are present, the mechanism needs not to evaluate a worker on all the tasks she solves at time t.

Parameter α is set as suggested in the corollary, and its proper value depends on the lowest allowable proficiency p_l, the characteristics of the considered dataset (the value of γ) and the quality of estimator $\mathcal{F}$ (the value of q).

Algorithm 1: PropeRBoost

```
**Data**: Initial reputation $\rho_0 > 0, p_l \in [\frac{1}{2}, 1), \eta \in (0, \frac{1}{2}]$
**begin**
    $\rho_w = \rho_0$;
    $\alpha = \frac{2\cdot(1-\gamma)\cdot(p_l-\frac{1}{2})\cdot(q-\frac{1}{2})}{1+2\cdot(1-\gamma)\cdot(p_l-\frac{1}{2})\cdot(q-\frac{1}{2})}$;
    **for** $t = 1$ *to* $t = T$ **do**
        Publish scale $\sigma_w = \frac{\rho_w}{\rho_w+1}$ to worker $w$;
        Assign worker $w$ to task $\tau$;
        Obtain answer $X$;
        Estimate the correct answer: $\hat{\theta}_{\mathcal{F}} = \mathcal{F}(\tau)$;
        Estimate performance:
        $1 : \hat{\pi}_w = \mathbb{1}_{X=\hat{\theta}_{\mathcal{F}}} - \Pr(\hat{\theta}_{\mathcal{F}} = X)$;
        $2 : score_w = (1-\alpha)\cdot\hat{\pi}_w - \alpha$;
        Update reputation: $\rho_w = \rho_w \cdot (1 + \eta \cdot score_w)$;
        Pay $\sigma_w \cdot Payment(X)$ to worker $w$;
```

Once we estimate worker w's performance, the reputation of the worker is updated in such a way that it exponentially increases or decreases depending on the worker's score. For example, if the worker is providing random reports, her score is likely to be negative, in which case her reputation decreases by a factor $1 + \eta \cdot score_w < 1$. On the other hand, if the worker performs well, her score is likely to be positive, in which case her reputation grows by factor $1 + \eta \cdot score_w > 1$. Clearly, if worker w performs well most of the time, her reputation ρ_w is quickly boosted to large values, which means that scales $\sigma_w = \frac{\rho_w}{\rho_w+1}$ become quickly close to the maximum (i.e., 1). On the other hand, if the worker does not perform well, she will quickly loose her reputation, and her payments will become very close to the minimum (i.e., 0).

The mechanism has initial reputation ρ_0 as its parameter: the initial reputation should be set to relatively small values, so that workers first prove themselves to the mechanisms by building up their reputation. The next section shows how the total gain of the mechanism is bounded from below by a negative value of the initial reputation. Notice, however, that the value of the initial reputation should not be too small because if workers cannot build up their reputation in a reasonable amount of time, they will not respond to payments.

The reputation update also depends on parameter η. This parameter should not exceed $\frac{1}{2}$, but its proper value depends on proficiency threshold p_u, as well as the variance of $score_w$. As it is shown in the following subsections, a good value for η would be $\eta = \min(\frac{1}{2}, A \cdot (p_u - p_l))$, where A is defined in Corollary 1. However, often for high quality reports, the expected score $\mathbb{E}(score_w)$ is greater than variance $Var(score_w)$, in which case, one can set $\eta = \frac{1}{2}$. This will typically be true if workers solve several tasks in each period t, and thus, worker w's reputation is updated using the average of scores $score_w$ across these tasks.

Finally, at the end of the period t, worker is rewarded with $\sigma_w \cdot Payment(X)$. Function $Payment$ can be any proper payment function that incentivizes workers to truthfully reveal their private information. If workers only strategize on whether they will invest high effort to obtain proficiency greater than p_u or low effort and have proficiency lower than p_l, one can use a constant payment function, and in this case, workers' payoffs can be known to them upfront (before they engage in solving the tasks).

PropeRBoost algorithm has several properties that makes it practical for crowdsourcing scenarios. These properties are described in the following subsections. To make our analysis easier to follow, we will denote time dependent variables by putting the subscript t. For example, worker w's reputation at time t is denoted by $\rho_{w,t}$.

Property 1: Bounded Negative Gain

We first investigate the impact of low quality workers on the utility of the mechanism, described by equation (3). The first property we show is that no matter what worker w's strategy is, the overall expected gain of the mechanism is bounded from below by a constant proportional to the initial reputation ρ_0. By noting that scaling with $\sigma = 0$ always brings 0 gain, we directly obtain that, in expectation, PropeRBoost is never much worse than having no rewarding system at all. More formally, we have the following result:

Theorem 1. *The minimal value of the PropeRBoost's total gain is bounded from below by:*

$$\mathbb{E}(\sum_{t=1}^{T} g(\sigma_{w,t}, p_t)) \geq -\frac{\rho_0}{A \cdot \eta}$$

where $\sigma_{w,t}$ denotes the scale assigned to worker w at time t, p_t is the worker's proficiency at time t, and A is defined in Corollary 1.

Proof. We have:

$$\begin{aligned}
\ln(\rho_{w,T+1} + 1) &= \ln(\rho_{w,T} \cdot (1 + \eta \cdot score_{w,T}) + 1) \\
&= \ln((\rho_{w,T} + 1)(1 + \frac{\rho_{w,T}}{\rho_{w,T} + 1} \cdot \eta \cdot score_{w,T}))) \\
&= \ln(\rho_{w,T} + 1) + \log(1 + \sigma_{w,T} \cdot \eta \cdot score_{w,T}) = ...
\end{aligned}$$

$$\begin{aligned}
&= \ln(\rho_0 + 1) + \sum_{t=1}^{T} \log(1 + \sigma_{w,t} \cdot \eta \cdot score_{w,t}) \\
&\leq \ln(\rho_0 + 1) + \eta \cdot \sum_{t=1}^{T} \sigma_{w,t} \cdot score_{w,t} \\
&\leq \rho_0 + \eta \cdot \sum_{t=1}^{T} \sigma_{w,t} \cdot score_{w,t}
\end{aligned}$$

where we used the fact that $\ln(1+x) \leq x$ for $x > -1$. From the way PropeRBoost updates the reputations, we know that $\rho_{w,t} > 0$, because $\rho_0 > 0$ and $1 + \eta \cdot score_w > 0$. The latter inequality follows from $\eta \leq \frac{1}{2}$ and $score_w \in [-1, 1]$. In the worst case, reputation $\rho_{w,T+1}$ is approximately equal to 0, so by setting $\rho_{s,T+1} = 0$, we obtain:

$$\sum_{t=1}^{T} \sigma_{w,t} \cdot score_{w,t} \geq -\frac{\rho_0}{\eta}$$

The expected gain $g(\sigma_w, p_t)$ at time t is:

$$\mathbb{E}(g(\sigma_{w,t}, p_t)) = \int_0^1 p(\sigma_{w,t} = \sigma) \cdot \sigma \cdot (p_t - p_l) d\sigma$$
$$= \int_0^1 p(\sigma_{w,t} = \sigma) \cdot \sigma \cdot \frac{\mathbb{E}(score_{w,t})}{A} d\sigma$$

where we applied $\mathbb{E}(score_{w,t}) = A \cdot (p_t - p_l)$ from Corollary 1.[1] Therefore, the obtained expression evaluates to $\frac{1}{A} \cdot \mathbb{E}(\sigma_{w,t} \cdot score_{w,t})$. This implies that the expected total gain $\sum_{t=1}^{T} g(\sigma_{w,t}, p_t)$ is equal to:

$$\mathbb{E}(\sum_{t=1}^{T} g(\sigma_{w,t}, p_t)) = \sum_{t=1}^{T} \mathbb{E}(g(\sigma_{w,t}, p_t))$$
$$= \frac{1}{A} \sum_{t=1}^{T} \mathbb{E}(\sigma_{w,t} \cdot score_{w,t}) = \frac{1}{A} \cdot \mathbb{E}(\sum_{t=1}^{T} \sigma_{w,t} \cdot score_{w,t})$$
$$\geq -\frac{1}{A} \cdot \frac{\rho_0}{\eta}$$

□

The importance of this theorem is that we can make the total negative gain of the mechanism arbitrarily small by putting the initial reputation of worker w to small enough value. This is true regardless of worker w's strategy, as long as estimator $\mathcal{F}$ performs better than random estimation. We have not discussed, however, the incentive properties of the mechanism, which might play a crucial role for the result to be valid. Namely, when $\mathcal{F}$ outputs estimates using the report of peer workers instead of gold tasks, to guarantee that the estimates are accurate, at least some portion of the peer workers should provide accurate reports. In other words, the mechanism should incentivize workers to provide accurate reports.

Property 2: Incentives for Accurate Reporting

The second property we want to show is that a worker whose performance is in expectation good and stable is able to quickly build up her reputation so that her payments quickly become close to the maximum. In particular, with the reputation updating rule defined by PropeRBoost algorithm, a worker w whose proficiency is expected to be $p > p_u$ should receive near maximal payments for honest reporting. That is, the scale assign to the worker should be close to 1 and approach 1 as the worker solves more tasks.

Theorem 2. *Suppose that an honest worker w has proficiency $p > p_u$ (in all time periods t), and let parameter η be strictly greater than 0 and less than:*

$$\begin{cases} \frac{1}{2} & \text{if } Var(score_{w,t}) < \mathbb{E}(score_{w,t}) \\ \min(\frac{1}{2}, A \cdot (p_u - p_l)) & \text{if } Var(score_{w,t}) \geq \mathbb{E}(score_{w,t}) \end{cases}$$

where A is defined in Corollary 1 and $Var(score_w)$ is the variance of the score $score_{w,t}$. Then, the expected difference between the optimal scale for this worker ($\sigma = 1$) and scales $\sigma_{w,t}$ generated by PropeRBoost is bounded by:

$$\mathbb{E}(\sum_{t=1}^{T}(1 - \sigma_{w,t})) \leq \left[\frac{2}{h} \cdot \ln(\frac{\rho_0 + 1}{\rho_0}) + \frac{e^{-\frac{1}{4} \cdot h^2}}{1 - e^{-\frac{1}{4} \cdot h^2}}\right]$$

where $h = \mathbb{E}(\ln(1 + \eta \cdot score_{w,t})) > 0$.

Proof. Markov's inequality gives us:

$$\mathbb{E}(\sigma_{w,t}) = \mathbb{E}(\frac{\rho_{w,t}}{\rho_{w,t} + 1}) \geq \Pr(\rho_{w,t} \geq \rho_0 \cdot a_t) \cdot \frac{\rho_0 \cdot a_t}{\rho_0 \cdot a_t + 1}$$

where we used: $a_t = e^{\frac{1}{2} \cdot h \cdot t}$, where $h = \mathbb{E}(\ln(1 + \eta \cdot score_{w,t})) > 0$. Notice that $score_{w,t} \in [-1, 1]$. Therefore, from $\ln(1 + x) \geq x - x^2$ for $x \geq -\frac{1}{2}$, it follows that $h \geq \eta \cdot \mathbb{E}(score_{w,t}) - \eta^2 \cdot \mathbb{E}((score_{w,t})^2)$. Due to the conditions of the theorem, we know that $\eta < \mathbb{E}(score_{w,t})$ or $\mathbb{E}(score_{w,t}) > \frac{1}{2} \cdot \mathbb{E}((score_{w,t})^2)$ (when $Var(score_{w,t}) < \mathbb{E}(score_{w,t})$), which by $score_{w,t} \in [-1, 1]$, implies that $h > 0$. Now, notice that:

$$\Pr(\rho_{w,t} \geq \rho_0 \cdot a_t) = \Pr(\ln \rho_{w,t} \geq \ln(\rho_0 \cdot a_t))$$
$$= \Pr(\ln \rho_{w,t} \geq \ln \rho_0 + \frac{1}{2} \cdot h \cdot t)$$
$$= \Pr(\ln \rho_{w,t} - t \cdot h - \ln \rho_0 \geq -\frac{1}{2} \cdot h \cdot t)$$
$$\geq 1 - \Pr(\ln \rho_{w,t} - t \cdot h - \ln \rho_0 \leq -\frac{1}{2} \cdot h \cdot t)$$
$$= 1 - b_t$$

where we set $b_t = \Pr(\ln \rho_{w,t} - t \cdot h - \ln \rho_0 \leq -\frac{1}{2} \cdot h \cdot t)$. Since $\ln \rho_{w,t} - \ln \rho_0$ is a sum of t independent random variables $\ln(1 + \eta \cdot score_{w,t})$ that are in expectation equal to h, using Hoeffding's inequality, we obtain:

$$b_t \leq e^{-\frac{2 \cdot (h \cdot t)^2}{4 \cdot \sum_{\tau=1}^{t} [\ln \frac{3}{2} - \ln \frac{1}{2}]^2}} \leq e^{-\frac{1}{4} \cdot h^2 \cdot t}$$

Therefore, $\mathbb{E}(\sigma_{w,t})$ is bounded from below by:

$$\mathbb{E}(\frac{\rho_{w,t}}{\rho_{w,t} + 1}) \geq (1 - b_t) \cdot \frac{\rho_0 \cdot a_t}{\rho_0 \cdot a_t + 1}$$
$$\geq 1 - \left[\frac{1}{\rho_0 \cdot a_t + 1} + e^{-\frac{1}{4} \cdot h^2 \cdot t} \cdot \frac{\rho_0 \cdot a_t}{\rho_0 \cdot a_t + 1}\right]$$

which means that $\mathbb{E}(\sum_{t=1}^{T}(1 - \sigma_{w,t}))$ is at most:

$$\sum_{t=1}^{T} \frac{1}{\rho_0 \cdot a_t + 1} + \sum_{t=1}^{T} e^{-\frac{1}{4} \cdot h^2 \cdot t} \cdot \frac{\rho_0 \cdot a_t}{\rho_0 \cdot a_t + 1}$$

[1] Notice that $\mathbb{E}(score_{w,t})$ is conditioned on $\sigma_{w,t}$ if worker w's strategy (proficiency p_t) is dependent on her reputation.

Let us put the bound for each term in the bracket. Using the fact that $a_t = e^{\frac{1}{2}\cdot t\cdot h}$, we obtain:

$$\sum_{t=1}^{T} \frac{1}{\rho_0 \cdot a_t + 1} = \sum_{t=1}^{T} \frac{1}{\rho_0 \cdot e^{\frac{1}{2}\cdot t\cdot h} + 1}$$
$$\leq \int_{t=0}^{T} \frac{1}{\rho_0 \cdot e^{\frac{1}{2}\cdot t\cdot h} + 1} dt \leq \int_{t=0}^{\infty} \frac{1}{\rho_0 \cdot e^{\frac{1}{2}\cdot t\cdot h} + 1} dt$$
$$= \frac{2}{h} \cdot \ln(\frac{\rho_0 + 1}{\rho_0})$$

Furthermore, we have:

$$\sum_{t=1}^{T} e^{-\frac{1}{4}\cdot h^2 \cdot t} \cdot \frac{\rho_0 \cdot a_t}{\rho_0 \cdot a_t + 1} \leq \sum_{t=1}^{T} e^{-\frac{1}{4}\cdot h^2 \cdot t}$$
$$= e^{-\frac{1}{4}\cdot h^2} \cdot \sum_{t=0}^{T-1} e^{-\frac{1}{4}\cdot h^2 \cdot t} \leq e^{-\frac{1}{4}\cdot h^2} \cdot \sum_{t=0}^{\infty} e^{-\frac{1}{4}\cdot h^2 \cdot t}$$
$$= \frac{e^{-\frac{1}{4}\cdot h^2}}{1 - e^{-\frac{1}{4}\cdot h^2}}$$

where we applied $\sum_{t=0}^{\infty} x^t = \frac{1}{1-x}$ for $x \in (0, 1)$. □

The bound in Theorem 2 implies that a worker with a good performance (i.e., $p > p_u$) will have in expectation average payment that is close to maximum (i.e., payment that the worker would obtain with the scale equal to 1). Notice that the average difference between the maximum payment and the one that the worker obtains is inversely proportional to time T, with a proportionality constant that depends on $h = \mathbb{E}(\ln(1 + \eta \cdot score_{w,t}))$: the higher h is, the lower the constant is. Clearly, worker w can increase the value of h with her performance, and thus, lower the proportionality constant.

Furthermore, the mechanism itself can influence the value of h by adjusting parameter η. Notice that parameter η can always be set to $\min(\frac{1}{2}, A \cdot (p_u - p_l))$ for the theorem to hold. From Theorem 2, it follows that threshold p_u determines the lower bound for the proficiency levels that are guaranteed to achieve near-maximum scaling factors. While one might want to put p_u close to p_l, this also decreases the value of η, and thus, the value of h. Clearly there is a tradeoff between the thresholds' separation and the efficiency of the mechanism. However, high quality workers might have stable scores, meaning that $Var(score_{w,t})$ is lower than $E(score_{w,t})$ for good proficiency levels. For example, workers might solve several micro-tasks at each time step t, which significantly reduces the variability in their performance scores. In this case, one can set η to any value in $(0, \frac{1}{2}]$, as indicated by Theorem 2.

Notice that Theorem 2 does not examine the possibility that a worker w could manipulate the system; it only provides guarantees that an honest worker with a desirable proficiency level is expected to have an average payment close to the one with the scale equal to 1.

Property 3: Near-Minimal Payments for Low Quality Reports

Property 2 shows that workers are rewarded for high quality reports, but we must also ensure that alternative strategies, such as random reporting, lead to small average rewards. We show that, in expectation, strategies for which a worker's proficiency is strictly smaller than $p < p_l$ lead to very low average payments, close to 0 in a long run. Let us consider first what happens when worker w consistently reports low quality information.

Proposition 2. *Consider a worker w that has proficiency $p < p_l$ (in all time periods t) and a payment function $Payment(X)$ that takes values in $[0, B]$. Then, the expected payment to worker w over time period T is bounded from above by:*

$$\mathbb{E}(\sum_{t=1}^{T} \sigma_{w,t} \cdot Payment(X_t)) \leq B \cdot \frac{\rho_0}{A \cdot \eta \cdot (p_l - p)} \quad (4)$$

where A is defined in Corollary 1.

Proof. The expected payment at time t to a worker whose w is:

$$\mathbb{E}(\sigma_{w,t} \cdot Payment(X_t)) \leq B \cdot \mathbb{E}(\frac{\rho_{w,t}}{\rho_{w,t} + 1})$$
$$\leq B \cdot \mathbb{E}(\rho_{w,t})$$

where the inequalities are due to $Payment(X_t) \in [0, B]$ and $\rho_{w,t} > 0$. Furthermore, we have:

$$\mathbb{E}(\rho_{w,t}) = \mathbb{E}(\rho_0 \prod_{\tau=1}^{t} (1 + \eta \cdot score_{w,\tau}))$$

Due to the independence of scores $score_{w,t}$ and the fact that $\mathbb{E}(score_{w,t}) = A \cdot (p - p_l)$, we obtain:

$$\mathbb{E}(\rho_0 \prod_{\tau=1}^{t} (1 + \eta \cdot score_{w,\tau})) = \rho_0 \cdot (1 + \eta \cdot A \cdot (p - p_l))^t$$

Therefore, the total expected payment to a worker w is bounded from above by:

$$\mathbb{E}(\sum_{t=1}^{T} \sigma_{w,t} \cdot Payment(X_t))$$
$$\leq B \cdot \sum_{t=1}^{T} \rho_0 \cdot (1 + \eta \cdot A \cdot (p - p_l))^t$$
$$\leq B \cdot \sum_{t=0}^{\infty} \rho_0 \cdot (1 + \eta \cdot A \cdot (p - p_l))^t$$
$$= B \cdot \frac{\rho_0}{A \cdot \eta \cdot (p_l - p)}$$

where we used the fact that $p - p_l < 0$ and $\sum_{t=0}^{\infty} x^t = \frac{1}{1-x}$ for $0 \leq x < 1$. □

The direct consequence of the proposition is that the average payoff of a low quality worker is in expectation close to 0 in a long run, and the result follows by dividing bound (4) with period T. Notice that the bound in the proposition is inversely proportional to $p_l - p > 0$, which means that the worse the worker is, the quicker her average payoff will approach 0.

Proposition 2 assumes that worker w's strategy is independent of time. To generalize it, we investigate the situation where a worker is allowed to base her strategy on both time and the current value of her reputation score. For example, a worker might decide to invest high effort and be more accurate if her reputation ρ_w is smaller than 5, and invest low effort and be less accurate if her reputation is greater than 5. We show that no matter what worker w's strategy is, her average payoff over a longer period will be equal to 0 if her average proficiency converges to values smaller than p_l. This effectively discourages any spamming strategy that tends to provide low quality reports.

Theorem 3. *Consider a payment function* $Payment(X)$ *that takes values in* $[0, B]$ *and a worker* w *whose average proficiency,* $\hat{p}_T = \frac{1}{T}\sum_{t=1}^{T} p_t$*, converges to a value* $\lim_{T\to\infty} \hat{p}_T = \hat{p} < p_l$*. Then, the expected value of the average payment to worker* w *converges to* 0*, i.e.:*

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}(\sigma_{w,t} \cdot Payment(X_t)) = 0$$

Proof. As shown at the beginning of the proof of Proposition 2, the expected payment at time t is bounded by $B \cdot \mathbb{E}(\sigma_{w,t})$.

Now, since $\lim_{T\to\infty} \hat{p}_T = \hat{p}$, we know that for any $\epsilon > 0$, there exists $T_0 \geq 1$ such that for any $T' > T_0$ we have $p_t < \hat{p} + \epsilon$. Let us put $\epsilon = \frac{p_l - \hat{p}}{2}$, which gives us $p_t < \frac{p_l + \hat{p}}{2}$. For $T > T_0$, we have :

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}(\sigma_{w,t} \cdot Payment(X_t)) \leq B \cdot \frac{T_0}{T} + B \cdot \frac{T - T_0}{T} \sum_{t=T_0}^{T} \mathbb{E}(\sigma_{w,t}) \qquad (5)$$

Using the same approach as in Proposition 2, we obtain that $\mathbb{E}(\sigma_{w,t})$ is, for $t > T_0$, equal to:

$$\mathbb{E}(\sigma_{w,t}) \leq \mathbb{E}(\rho_{T_0,w}) \cdot \prod_{\tau=T_0}^{t} (1 + \eta \cdot score_{w,\tau}) \leq \mathbb{E}(\rho_{T_0,w}) \cdot (1 - \frac{1}{2} \cdot \eta \cdot A \cdot (p_l - \hat{p}))^{t - T0}$$

where we used the fact that $p_t < \frac{p_l + \hat{p}}{2}$. Therefore:

$$\sum_{t=T_0}^{T} \mathbb{E}(\sigma_{w,t}) \leq \mathbb{E}(\rho_{T_0,w}) \sum_{t=T_0}^{\infty} (1 - \frac{\eta \cdot A}{2} \cdot (p_l - \hat{p}))^{t - T0} \leq \frac{2 \cdot \mathbb{E}(\rho_{T_0,w})}{\eta \cdot A \cdot (p_l - \hat{p})}$$

Since T_0 and $\mathbb{E}(\rho_{T_0,w})$ are bounded from above, by letting $T \to \infty$, we obtain that expression (5) goes to 0. □

Simulations

We consider two simulation scenarios: (1) a multi-task crowdsourcing setting in which workers solve a bundle of tasks at each time step and are rewarded using a peer consistency approach; (2) a crowd-sensing setting where a crowdsensor reports one measurement at each time step and is rewarded with a payment rule that compares its report to a report of a trusted sensor.

Multi-task Crowdsourcing

In the first scenario, we consider a synthetic dataset in which each task has an underlying true binary answer, a or b, with the prior probability of an answer set to a generic value: $Pr(\theta = a) = 0.4$ and $Pr(\theta = b) = 0.6$. Apart from worker w, whose responses we analyze, there are another 50 workers whose proficiencies are generated uniformly at random from the interval $[0.5, 1]$. Each worker is assigned with 12 randomly chosen tasks in a bundle of tasks containing 100 tasks. These values are set for all bundles of tasks, in total 500 of them (i.e., $T = 500$ from a worker w's point of view).

We examine the properties of the PropeRBoost payments with respect to a state of the art peer consistency mechanism introduced in (Dasgupta and Ghosh 2013), which we scale so that the payments take values in interval $[0, 1]$.[2] To estimate the performance of worker w at each time step, the PropeRBoost algorithm uses expression (1), where $\Pr(\hat{\theta}_{\mathcal{F}} = X)$ is estimated by randomly sampling a task not solved by worker w and calculating $\mathbb{1}_{\hat{\theta}_{\mathcal{F}} = X}$ — in expectation this expression is equal to prior $\Pr(\hat{\theta}_{\mathcal{F}} = X)$. Furthermore, the implementation of the PropeRBoost's estimator $\mathcal{F}$ is defined by the algorithm from (Karger, Oh, and Shah 2011), which has provable bounds on the quality of estimations. Initial value of reputation ρ_w is set to $\rho_0 = 0.1$, while α is set using the expressions in Proposition 1 and Corollary 1, with q estimated from the data ($q \approx 0.9$) and proficiency threshold p_l set to $p_l = 0.7$. Parameter η is equal to $\eta = 0.25$.

Let us first examine the payments provided to a worker w by the baseline algorithm for two basic strategies: *random* - defined by proficiency $p = 0.6$; *honest* - defined by proficiency $p = 0.85$. These payments are shown in Figure 1 for time periods $t \in \{1, ..., T\}$. The difference in payments between the two strategies is relatively small; closer inspection reveals that, on average, the payments for honesty are larger than for random reporting by about 0.09. If we take into account that the payments have large standard deviations for both strategies (around 0.07), it is unlikely that a worker would notice any difference in payments for the two strategies because the payments are received in an online manner (after solving each bundle of tasks). By taking into account that a worker experiences higher cost of effort for honesty than for spamming, random reporting seems more profitable. Furthermore, the payments for random reporting do not decrease over time, meaning that the mechanism gives away a large amount of monetary rewards in return for random data.

[2] Since we randomly assign the tasks, it can happen that a worker w has no peers for certain tasks, in which case the payments are set to 0.5.

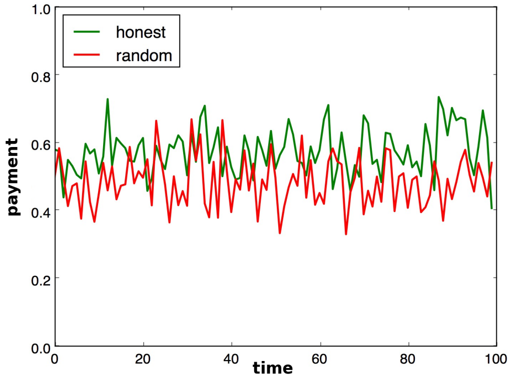

Figure 1: Payments for different strategies calculated by the mechanism of (Dasgupta and Ghosh 2013) (every 5th step is shown).

Let us now examine the scales for a worker w that the PropeRBoost algorithm outputs for strategies: *random*; *honest*; *switch* - reporting honestly for the first half of the reporting period and reporting randomly in the second half of the period; *keepRep* - reporting honestly when reputation ρ_w is lower than the initial reputation ρ_0, while reporting randomly when reputation ρ_w is above its initial value ρ_0. The evolution of scales over time is shown in Figure 2. The scale for random reporting quickly converges to 0. On the other hand, the scale for honesty converges to 1. In other words, the difference between the average payments for honesty and random reporting is maximal in a long run. If the worker switches her strategy from honest to random, i.e., she uses the *switch* strategy, the area below the switch curve in the first half of the reporting period is approximately equal to the area below the switch curve in the second half of the reporting period, indicating that the total utility of PropeRBoost is not overly negative (see Theorem 1). A more sophisticated reporting strategy is the *keepRep* strategy, where a worker tries to maintain the same scale using the least effort as possible. The average level of proficiency is still greater than minimal acceptable level p_l, i.e., it is equal to 0.71.

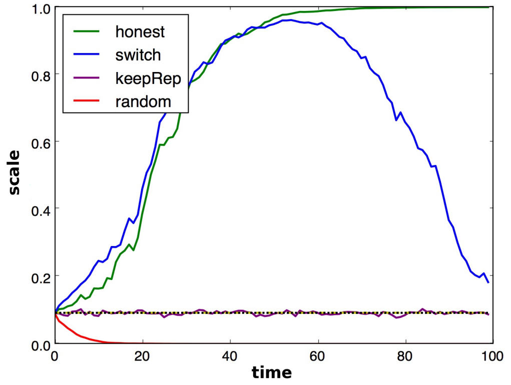

Figure 2: PropeRBoost scales for different strategies (every 5th step is shown). Black and yellow dashed line represents the initial scale.

Finally, we show the evolution of the payments of PropeRBoost when the payment function is the mechanism of (Dasgupta and Ghosh 2013). By comparing Figure 3 with Figure 1, one can easily see the effectiveness of the PropeRBoost algorithm - it provides much clearer separation between rewards for the two different strategies than the traditional scaling approach does.

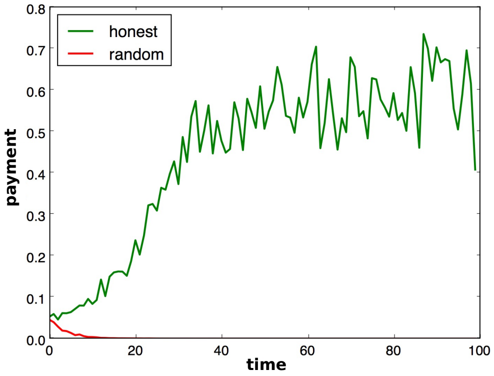

Figure 3: PropeRBoost with the payment mechanism of (Dasgupta and Ghosh 2013) for different strategies (every 5th step is shown).

Crowd-sensing

Our crowd-sensing test-bed is based on a real dataset containing levels of UFP (ultra fine particles) over Zurich urban area — the full description of the dataset can be found in (Hasenfratz et al. 2014) — and a region based Gaussian Processes model that incorporates the spatial features of the terrain, as described in (Jutzeler, Li, and Faltings 2014). One chunk of data contains aggregated measurements over the period of two weeks for 200 locations. In total, we have 22 chunks of data, 2 for each of 11 different months. Half of the data is used for training the GP model and the other half for testing purposes. To simulate finer-grained periods of time, e.g., hourly sensing periods, we iterate through the dataset 800 times and generate for each test value in the data, 800 artificial measurement. These measurements are obtained by sampling from a normal distribution with a mean equal to the test value and a standard deviation equal to the half of the standard deviation of the prior (the prior is obtained from the GP model). In total, we have 8800 periods, which, if the base period is 1 hour, corresponds to approximately 1 year.

We score crowd sensor w using the measurement of a trusted sensor, which measures at a different location. The locations of the two sensors are randomly chosen at each time period t from 200 available locations. To measure performance $\hat{\pi}_w$, we convert the measurement of sensor w into a binary signal using a simple rule: if the measurement is greater than the mean of the prior at the location of sensor w, then the binary signal is equal to 1; and otherwise is 0. We do the same conversion for the mean of the posterior at the location of w calculated using the measurement of the trusted sensor and the GP model (this defines the binary signal of the trusted sensor). The binary signal of sensor w

represents X, the binary signal of the trusted sensor represents $\hat{\theta}_{\mathcal{F}}$, while the noiseless measurement of w represents θ. From the data we estimate the accuracy of $\hat{\theta}_{\mathcal{F}}$ ($q \approx 0.6$) and the proficiency of a good sensor ($p \approx 0.86$). Since the mean of the prior is used as a threshold in the binary conversion, we know that $Pr(\theta = 0) = Pr(\theta = 1) = 0.5$, which also implies $Pr(\hat{\theta}_{\mathcal{F}} = 0) \approx Pr(\hat{\theta}_{\mathcal{F}} = 1) \approx 0.5$ for q that is not substantially biased towards a particular value of θ. By defining $p_l = 0.75$ and $p_u = 0.85$, we obtain that a good value of α is approximately 0.05, and the good value of η is approximately 0.02 (see Theorem 2). The initial reputation is set to $\rho_0 = 0.1$.

As a baseline payment method, we consider the quadratic scoring rule (Gneiting and Raftery 2007) that takes values in $[0, 1]$ and has two input variables: (1) the measurement of a trusted sensor; (2) the posterior probability distribution function of what a trusted sensor reports given the measurement of sensor w. The latter is calculated with the GP model. Using the same set of strategies as in the previous section, we analyze the baseline and our boosting mechanism. Notice that the measurement in the random strategy is obtained by sampling from a normal distribution with the mean equal to the real measurement and the standard deviation that is 4 times larger than the standard deviation of the prior.

As shown in Figure 4, the payments of the quadratic scoring rule for strategies honest and random are almost indistinguishable: the mean of the payments for the honest strategy is equal to 0.58 and the standard deviation is equal to 0.09; for the random strategy, the mean is equal to 0.56 and the standard deviation is equal to 0.1.

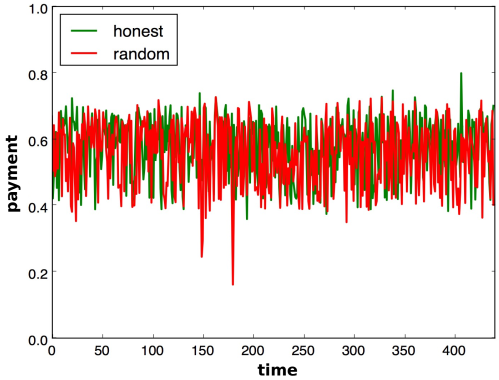

Figure 4: Payments for different strategies calculated using the quadratic scoring rule (every 20th step is shown).

Figure 5 shows how the scales of PropeRBoost evolve for different strategies. We obtain qualitatively the same results as in the previous section, with the keepRep strategy resulting in the average proficiency equal to $0.81 > p_l$. Figure 6 confirms that using PropeRBoost in crowd-sensing can lead to significantly more separated payments between honest and random reporting than the traditional approach does.

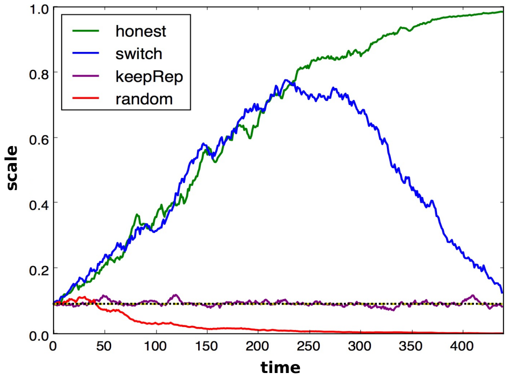

Figure 5: PropeRBoost scales for different strategies (every 20th step is shown). Black and yellow dashed line represents the initial scale.

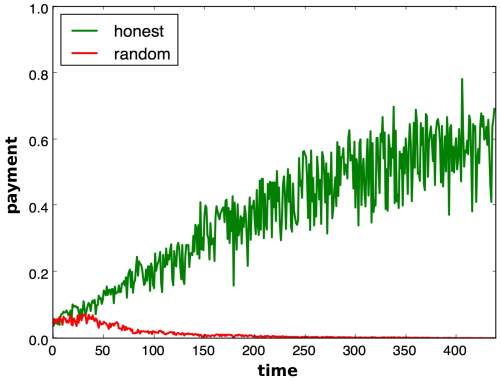

Figure 6: PropeRBoost with the quadratic scoring rule for different strategies (every 20th step is shown).

Conclusion

This paper investigates the problem of incentivizing workers who repeatedly interact with a crowdsourcing system. We have designed a novel boosting mechanism, PropeRBoost, that improves existing incentive schemes in terms of the differentiation between low and high quality performance. The main property of the mechanism is that it can provide workers with positive payments so that: 1) good workers receive average payments that are close to the maximum, i.e., close to the average with the maximum scale; 2) spammers receive near minimal payments. We consider these two properties to be of a great significance when it comes to understanding the reward system: workers are more likely to respond to an incentive when it clearly demonstrates its effectiveness.

The most challenging future step of this work is to experimentally compare the effectiveness of PropeRBoost with the effectiveness of the existing incentive mechanisms. While most of the existing mechanisms are designed for a single shot scenario, PropeRBoost relies on the repeated interactions of workers with the mechanism, making experimentation a more significant challenge.

Acknowledgments

The work reported in this paper was supported by Nano-Tera.ch as part of the OpenSense2 project. We thank Arnaud Jutzeler and Jason J. Li for providing a crowd-sensing testbed and the anonymous reviewers for useful comments and feedback.

References

Cesa-Bianchi, N., and Lugosi, G. 2006. *Prediction, Learning, and Games*. New York, NY, USA: Cambridge University Press.

Dasgupta, A., and Ghosh, A. 2013. Crowdsourced judgement elicitation with endogenous proficiency. In *Proceedings of the 22nd ACM International World Wide Web Conference (WWW'13)*.

Gneiting, T., and Raftery, A. E. 2007. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association* 102:359–378.

Harris, C. 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)*.

Hasenfratz, D.; Saukh, O.; Walser, C.; Hueglin, C.; Fierz, M.; and Thiele, L. 2014. Pushing the spatio-temporal resolution limit of urban air pollution maps. In *Proceedings of the 12th International Conference on Pervasive Computing and Communications (PerCom 2014)*.

Ho, C.-J.; Slivkins, A.; and Vaughan, J. W. 2014. Adaptive contract design for crowdsourcing markets: Bandit algorithms for repeated principal-agent problems. In *Proceedings of the Fifteenth ACM Conference on Economics and Computation*, EC '14.

Jutzeler, A.; Li, J. J.; and Faltings, B. 2014. A region-based model for estimating urban air pollution. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14.

Karger, D. R.; Oh, S.; and Shah, D. 2011. Iterative learning for reliable crowdsourcing systems. In *Proceeedings of the 24th Annual Conference on Neural Information ProcessingSystems (NIPS'11)*.

Karger, D. R.; Oh, S.; and Shah, D. 2013. Efficient crowdsourcing for multi-class labeling. In *Proceedings of the ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS'13.

Levin, J. 2003. Implicit contracts, incentive compatibility, and involuntary unemployment. *American Economic Review* 93:835–857.

Liu, Q.; Peng, J.; and Ihler, A. T. 2012. Variational inference for crowdsourcing. In *Proceeedings of the 25th Annual Conference on Neural Information ProcessingSystems (NIPS'12)*.

MacLeod, B., and Malcomson, J. 1989. Implicit contracts, incentive compatibility, and involuntary unemployment. *Journal of the Econometric Society* 57:447–480.

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

Nisan, N.; Roughgarden, T.; Tardos, E.; and Vazirani, V. V. 2007. *Algorithmic Game Theory*. New York, NY, USA: Cambridge University Press.

Oleson, D.; Sorokin, A.; Laughlin, G.; Hester, V.; Le, J.; and Biewald, L. 2011. Programmatic gold: Targeted and scalable quality assurance in crowdsourcing. In *Proceedings of the AAAI Workshop on Human Computation*.

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

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

Radanovic, G., and Faltings, B. 2016. Limiting the influence of low quality information in community sensing. In *Proceedings of the 2016 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS'16.

Radanovic, G.; Faltings, B.; and Jurca, R. 2016. Incentives for effort in crowdsourcing using the peer truth serum. *ACM Transactions on Intelligent Systems and Technology* 7:48:1–48:28.

Raykar, V. C.; Yu, S.; Zhao, L. H.; Valadez, G. H.; Florin, C.; Bogoni, L.; and Moy, L. 2010. Learning from crowds. *J. Mach. Learn. Res.* 11:1297–1322.

Resnick, P., and Sami, R. 2007. The influence limiter: Provably manipulation-resistant recommender systems. In *Proceedings of ACM RecSys 07 Conference on Recommender Systems*, 25–32.

Shah, N. B., and Zhou, D. 2015. Double or nothing: Multiplicative incentive mechanisms for crowdsourcing. In *Advances in Neural Information Processing Systems*, NIPS'15.

Shah, N.; Zhou, D.; and Peres, Y. 2015. Approval voting and incentives in crowdsourcing. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*. JMLR Workshop and Conference Proceedings.

Shnayder, V.; Agarwal, A.; Frongillo, R.; and Parkes, D. 2016. Informed truthfulness in multi-task peer prediction. In *Proceedings of the Seventeenth ACM Conference on Economics and Computation (To appear)*, EC '16.

Soufani, H. A.; Parkes, D. C.; and Xia, L. 2012. Random utility theory for social choice. In *Proceeedings of the 25th Annual Conference on Neural Information ProcessingSystems (NIPS'12)*.

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

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

Witkowski, J.; Bachrach, Y.; Key, P.; and Parkes., D. C. 2013. Dwelling on the negative: Incentivizing effort in peer prediction. In *Proceedings of the 1st AAAI Conference on Human Computation and Crowdsourcing*.