Incentives to Counter Bias in Human Computation

Boi Faltings and **Pearl Pu** and **Bao Duy Tran**
Swiss Federal Institute of Technology (EPFL)
1015 Lausanne, Switzerland
{boi.faltings|pearl.pu|baoduy.tran}@epfl.ch

Radu Jurca
Google
Zürich, Switzerland
radu.jurca@gmail.com

Abstract

In online labor platforms such as Amazon Mechanical Turk, a good strategy to obtain quality answers is to take aggregate answers submitted by multiple workers, exploiting the wisdom of the crowd. However, human computation is susceptible to systematic biases which cannot be corrected by using multiple workers. We investigate a game-theoretic bonus scheme, called Peer Truth Serum (PTS), to overcome this problem. We report on the design and outcomes of a set of experiments to validate this scheme. Results show Peer Truth Serum can indeed correct the biases and increase the answer accuracy by up to 80%.

Introduction

Online labor markets such as Amazon Mechanical Turk are widely used for problems that machine intelligence cannot yet solve, such as interpreting and labeling images or assessing subjective content. Problems are formulated as human intelligence tasks (HITs) and solved by several human workers recruited through the platform. For many tasks, especially subjective ones such as annotating an image with a set of keywords, the validity of the answer is hard to establish. This is problematic because workers have an incentive to minimize the effort they spend on the task, and tend to use heuristic problem solving strategies that involve little or no effort on the task itself. These include answering randomly, giving the answer that is considered most likely, or always giving the same answer. While it is common to filter out workers who systematically use heuristic strategies from their answers on "gold" tasks with known answers, this still leaves workers who resort to heuristic strategies only some of the time.

When the heuristic strategies result in random errors, accuracy can be increased by increasing the number of workers and aggregating their results, exploiting the wisdom of the crowd. However, this is not possible when the heuristic strategy has a common bias that can be mistaken for the true value. Such bias arises in many heuristic problem solving strategies, for example:

- when a worker has carried out several instances of a task that all had the same answer, and thus develops a bias towards that answer. For example, if the task is to classify review texts and the first 10 were all positive, if the 11th is negative it is easily overlooked.
- common beliefs, when there is a generally known most likely answer to the task. For example, when looking for grammatical errors in text, most sentences have no error so that is an obvious common answer.
- when a hint to a common answer is provided by the task itself that the workers use as a basis for bias. In this case, workers cannot escape the anchoring effect (Tversky and Kahneman, 1974) and will bias their answers towards this value. For example, if the task is to count the number of people in a crowded bus and the task description mentions that the bus has 60 seats, answers are likely to be close to that number. The effect is also observed when workers have access to the answers of other workers (Lorenz et al., 2011).

Figure 1: *A typical Mechanical Turk task: count the number of visible cameras/camcorders/binoculars/phones in the image.*

Increasing the number of workers is not effective for eliminating such biases. When there is a large set of similar tasks, it is possible to use statistical techniques to eliminate systematic bias (Ipeirotis et al., 2010). However, it is clearly

preferable to motivate workers to provide unbiased answers in the first place.

In this paper, we consider different bonus schemes and evaluate their influence on overcoming bias in workers' answers from a game-theoretic perspective. We propose in particular a novel incentive scheme, the *Peer Truth Serum*, that combines elements of the Bayesian Truth Serum (Prelec, 2004) with Peer Consistency (Huang and Fu, 2013).

Next, we present an experimental evaluation of the scheme on Amazon Mechnical Turk. As the subjective bias of individual workers can vary, we designed a task where bias is injected explicitly and uniformly by explicit priming of the worker with a likely answer. We show that this indeed introduces a bias in workers' answers that is difficult to correct with common methods for quality control. We show furthermore that the Peer Truth Serum bonus scheme corrects it. We believe that the results we obtain also hold lessons for other sources of bias.

Incentive mechanisms for crowdsourcing

The Mechanical Turk platform allows both negative incentives by rejecting workers' results, and positive incentives in the form of bonuses. Rejecting results is considered a very harsh measure, as workers can get excluded from the platform if their results are too frequently rejected. Thus, it should only be used in extreme cases. Bonus schemes are a more appropriate measure for incentivizing workers.

Harris (Harris, 2011) presented a study of incentives for screening resumes on Amazon Mechanical Turk. Workers were asked to evaluate on a scale of 1-5 the fit of a job application to a job description. The incentives were based on comparing the answers with the judgement of an experienced human resources expert; workers are either rewarded for agreeing or punished for disagreeing with an expert on the same task. The study found that incentives had a significant effect on accuracy and that the best effect was obtained by bonuses. Shaw, Horton and Chen (Shaw et al., 2010) tested a large variety of schemes using a task of classifying the type of content present on a web site. Their study found that schemes based on agreement were the most effective. The authors of the study concluded that workers think about the responses that others would give and thus work more objectively.

Giving rewards for agreeing with another worker has also been used by the very successful ESP game (von Ahn and Dabbish, 2004), where players are rewarded for assigning the same label as a peer to an image. More recently, (Huang and Fu, 2013) 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 also found that providing bonuses based on evaluating workers against randomly chosen peers increases accuracy, and even more so than comparing against a gold standard. The same authors also showed that social pressure can further increase accuracy (Huang and Fu, 2013).

(Dasgupta and Gosh, 2013) uses a version of Peer Consistency where rewards are constant and that requires that workers provide correct answers over 50% of the time. They note in particular that bonuses that incentivize truthfulness also incentivize effort and are thus doubly beneficial in a human computation context.

A major issue with Peer Consistency is that when the different answers have different likelihoods, it gives the highest rewards to answers that are the most common, since they are the most likely to match that of another worker. Thus, they encourage agents to report according to their biases. This effect was observed in the ESP game (von Ahn and Dabbish, 2004), where a heuristic strategy is to report very common labels that would more easily be matched by other players. Other studies of incentives based on agreement (Shaw et al., 2010; Huang and Fu, 2013; Huang and Fu, 2013) used tasks where answers had similar probabilities and thus did not involve bias.

To discourage the tendency to report common answers that are most likely to be matched by others, we should give higher rewards to surprising answers, provided that they are matched by other workers. In the ESP game, this was done by making common labels "taboo words" that give no points at all.

Game-theoretic analysis

A theoretical basis for such incentive schemes can be found in game theory, where the problem of rewarding agents to incentivize truthful reports of their private information has been extensively studied. We consider schemes where a requester provides a bonus that is scaled based on the information that is observed by the requester, i.e. the distribution of answers.

We first consider schemes that apply to a single individual task, i.e. the bonus for answer a_i within a vector of answers $\underline{\mathbf{a}}$ to the same task is

$$bonus(a_i) = f(a_i, \underline{\mathbf{a}})$$

The worker faces a choice between several strategies:

- cooperate: invest effort at most γ to find the true answer x and report it truthfully.
- deceive: invest effort to find the true answer x but report a different answer y.
- heuristic: do not invest any effort and report an answer according to a heuristic strategy.

A rational worker will evaluate the expected bonus for each strategy and pick the one for which she believes the bonus will be highest. As the bonus depends on the strategies simultaneously adopted by other agents, we have a game where rational agents follow equilibrium strategies. The bonus scheme must make *cooperate* a symmetric Nash equilibrium for all agents. We thus consider the best response under the assumption that all other workers cooperate.

The following proposition allows us to eliminate heuristic strategies from consideration:

Proposition 1 *Provided that the payoff of the best* cooperate *strategy is more than γ higher than the best* deceive *strategy, no* heuristic *strategy can be optimal.*

Proof 1 *The payoff of the best* heuristic *strategy cannot be better than that of the best* deceive *strategy by more than γ, since otherwise there would be a better* deceive *strategy that consists of investing effort and then applying the heuristic strategy. Thus, the best* cooperate *strategy is guaranteed to be better than the best* heuristic *strategy.*

Thus, we will only consider *cooperate* and *deceive* strategies below. Since both involve investing effort to solve the task, we will assume that the worker has solved the task and obtained an answer x and has to decide whether to report the answer truthfully or not.

As the worker does not know any of the other answers, the expected payoff depends crucially on the belief about the distribution of other workers' answers as embodied by $\underline{\mathbf{a}}$. We write $Pr(a_j)$ for the belief of a worker about another answer a_j before solving the task (the *prior*) and $Pr_x(a_j)$ for the belief after solving the the task and obtaining the result x (the *posterior*). We consider three types of posterior beliefs:

- self-dominant: x is the most likely value of another answer a_j:

$$(\forall y) Pr_x(a_j = x) \geq Pr_x(a_j = y) + \epsilon \tag{1}$$

- self-predicting: x is the value with the biggest increase in probability over a prior probability Pr:

$$\frac{Pr_x(x)}{Pr(x)} > \frac{Pr_x(y)}{Pr(y)} + \epsilon \tag{2}$$

- arbitrary: Pr_x can be an arbitrary distribution.

where ϵ is a gap that characterizes the confidence of the user in his answer.

Arbitrary beliefs It is easy to show by counter-example (see for example (Radanovic and Faltings, 2013)) that there is no function that makes cooperation a Nash equilibrium for arbitrary beliefs.

However, if the exact beliefs of workers are known, (Miller et al., 2005) have shown how to design a bonus function that is guaranteed to reward cooperation based on proper scoring rules, called the *peer prediction method*. (Jurca and Faltings, 2009) discuss variants and show how to avoid uninformative equilibria using a more complex function involving at least three of the other answers.

An important limitation of peer prediction methods is the need to know the agents' posterior probability distributions after each measurement. Zohar and Rosenschein ((Zohar and Rosenschein, 2006)) investigate mechanisms that are robust to variations of these distributions, and show that this is only possible in very limited ways and leads to large increases in payments. The Bayesian Truth Serum ((Prelec, 2004; Witkowski and Parkes, 2012b; Radanovic and Faltings, 2013)) is a mechanism that elicits both the estimate itself as well as the beliefs about other's estimates. This elicitation of extra information eliminates the need to know the prior beliefs, but also requires workers to report not only their answer, but also what they believe the *distribution* of other workers' answers to be. For example, for a task with 10 answers 1-10, a worker would have to not only select one of the answers but also 10 estimates of the fraction of workers giving the different answers. This information is used to explicitly reward answers that the worker considers unlikely, provided they are matched by a peer. As we will see in our experiment, requiring this extra information, even in simplified form, is perceived as difficult and unnatural.

A similar solution where agents report both their prior and posterior beliefs about the observed value is proposed in (Witkowski and Parkes, 2012a). Because Bayesian updating implies that the ratio of posterior/prior is the highest for the actually observed value, the two reports together also determine this observed value. However, it is difficult to apply this technique to crowdsourcing since we cannot enforce reporting the prior belief before executing a task, and it again would require workers to report more information than just the answer to the task.

To simplify the analysis for the following cases, we first consider a class of mechanisms that consists of selecting a random *peer* answer $a_j \in \underline{\mathbf{a}}, j \neq i$, and computing the bonus as a function of a_i and a_j. Furthermore, we use the insights gained in earlier experiments (Shaw et al., 2010) and consider reward schemes centered around *matching* answers.

Self-dominant beliefs For self-dominant, unknown beliefs, a very simple scheme is sufficient (Jurca and Faltings, 2003; Radanovic and Faltings, 2013) independently of the precise beliefs:

$$f(a_i, a_j) = \begin{cases} c & \text{if } a_i = a_j \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where c is a scaling constant chosen so that $c \cdot \epsilon \geq \gamma$, with ϵ the margin in the self-dominant condition 1 and γ the cost of effort.

This scheme encourages cooperation because by the self-dominant condition (1), the expected payoff of the *cooperate* strategy, $Pr_x(a_j = x) \cdot c$ exceeds that of any *deceive* strategy, $Pr_x(a_j = y) \cdot c$ by at least $c \cdot \epsilon \geq \gamma$. This is the game-theoretic justification for the Peer Consistency scheme (Shaw et al., 2010; Huang and Fu, 2013; Huang and Fu, 2013). For a wide range of crowdsourcing tasks, in particular those where answers do not have a very skewed distribution, worker beliefs are likely to satisfy the self-dominant assumption and so such a simple scheme is sufficient. Note that in contrast to mechanisms based on scoring rules, it does not require that agents' beliefs are known, nor that they are uniform across the population.

Self-predicting beliefs: the Peer Truth Serum When the distribution of answers is biased, and workers are not very sure about their own work, the self-dominant condition is often violated. For example, if the task is to detect animals in images, very few images contain cats, and a worker is not very sure if the cat she is seeing is actually there, then she may not believe that "cat" is indeed the most likely answer given by another worker.

However, in such as case the worker will still increase its belief that another worker also reports the same answer,

and can usually be expected this answer to have a higher increase in probability than other, correlated answers. This is expressed by the self-predicting condition 2.

For cases where this condition holds, we now define a simple incentive mechanism we call the *Peer Truth Serum*. It combines the intuitive simplicity of a reward based on agreement with another worker with the scaling of the reward according to the individual worker's beliefs that is achieved by the Bayesian Truth Serum.

To avoid the need for workers to also provide a separate prediction report, as required in the Bayesian Truth Serum, we use the public distribution of previous answers on other tasks (denoted as R) as a proxy for the unknown prior belief of the agent. If conveniently available, the answers of other peers can prime a worker and bias or replace her prior, uninformed expectations.

The mechanism we call the *Peer Truth Serum* pays a worker a bonus of $f(a_i, a_j, R)$ whenever her answer x matches that of another worker. The bonus is defined as:

$$f(a_i, a_j, R) = \begin{cases} \frac{c}{R(a_i)} & \text{if } a_i = a_j \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where c is a positive scaling constant. Such a scheme was first proposed for opinion polls in (Jurca and Faltings, 2011), for community sensing in (Faltings et al., 2013), and for a new form of peer prediction market in (Garcin and Faltings, 2014).

Proposition 2 *Whenever the agents' prior belief $Pr(x)$ is equal to the publicly available distribution $R(x)$, the Peer Truth Serum makes truthful reporting a Nash Equilibrium.*

Proof: Note that the expected reward for an agent who solves the task, obtains answer x and reports y is:

$$pay(x, y) = Pr_x(y) \cdot f(y, y, R)$$

The condition for solving the task and truthful reporting is being the best response by a margin greater than γ is:

$$\begin{aligned} \forall x, y, x \neq y : pay(x, x) - \gamma &> pay(x, y) \\ Pr_x(x) f(x, x, R) - \gamma &> Pr_x(y) f(y, y, R) \\ Pr_x(x) f(x, x, Pr) - \gamma &> Pr_x(y) f(y, y, Pr) \end{aligned}$$

where γ is the cost of effort for solving the task and obtaining answer x. If $f(x, x, R) = c/R(x)$ and $\gamma = c\epsilon$, the truthfulness condition is just the self-predicting condition 2. The scaling constant c has to be chosen in function of the margin ϵ that can be assumed in condition 2.

Note that this reward scheme has a very intuitive nature: it rewards answers that go against the biases expressed by $R(x)$, but on the other hand still requires matching another agent's answer so that only true answers would be considered. Thus, it is intuitive to understand and creates a natural tension between just reporting the most expected answer, and finding that the task at hand has a different correct answer with lower $R(x)$ and earning a higher bonus. Note further that it does not depend on the type of prior or posterior distributions, and works for any sample set that contains at least 2 answers (so for each answer there is at least one other against which it can be matched).

In contrast with scoring rules, this mechanism requires only that agents have the same prior biases (within some bounds), but does not require the posterior distributions to agree. The only condition is that the way a worker updates her beliefs respects Equation 2.

More general payment functions So far, we analyzed schemes that use a single, randomly selected peer answer. (Jurca and Faltings, 2009) have shown that any such scheme necessarily has at least one other, uninformative equilibrium where all agents always report the same value and that has a higher expected payoff than the truthful equilibrium. The same paper also showed how such equilibria can be eliminated by using not just one, but at least 3 peer answers, thus using the distribution of other answers rather than just a single one.

(Gao et al., 2014) report an empirical study on Mechanical Turk that shows that workers can learn to play such uninformative equilibria. However, the study concerned a game rather than a task requiring significant effort, and used a repeated game, a scenario that is far from normal human computation tasks.

Similarly, the Peer Truth Serum has an equilibrium strategy where all agents coordinate to report one of the least likely values. While in general coordination among crowd workers assigned to a particular task is difficult to achieve, it might happen where there is only a single very unlikely value.

This problem can be addressed by extending the mechanism from a single task to groups of similar tasks. The distribution R can then be taken from the answers to the group of tasks and does not have to be made available to the workers, thus making it impossible to coordinate on a specific value, similar to the mechanism proposed in (Dasgupta and Gosh, 2013). We describe such a mechanism in a forthcoming publication (Radanovic and Faltings, 2014b).

(Kamar and Horvitz, 2012) propose a modification of peer prediction for crowdsourcing where an answer is compared to the aggregate that would have been obtained without the agent present. However, no results on the performance are reported.

Experiment Design

To experiment with the effects of biased heuristic strategies, we require a task that is a typical human computation task and has two important features that were not present in tasks used in earlier studies: it should be tedious or subjective yet have a verifiable correct answer, and it must be possible to influence the bias that workers would use for a heuristic strategy so that uniform experimental conditions can be created.

We asked workers to count the number of visible cameras/camcorders/binoculars/phones in an image of a crowd, shown in Figure 1. This is a task that could be of interest for research in human behavior at public events, and we easily found many workers willing to do the task. At the same time, most people make a few mistakes in identifying individual

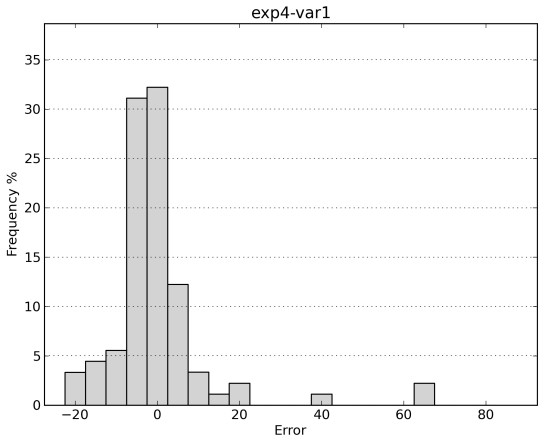

Figure 2: *Distribution of answers without bonus and without priming.*

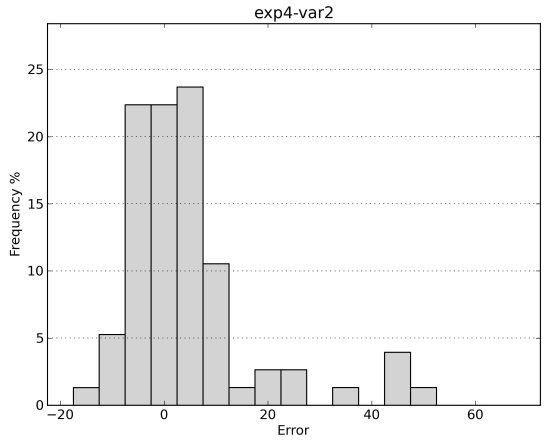

Figure 3: *Distribution of answers with priming to a prior expection of 60.*

items. As these add up linearly in the final count, we can expect errors to be normally distributed and thus the mean is the best estimator of the true value, which for this image is 34.

An important feature was that workers can safely be assumed to have no prior expectations about the task itself, so that we could easily set a bias manipulation ourselves. In order to control the bias conditions across the population, we took advantage of the anchoring effect (Tversky and Kahneman, 1974) and primed workers explicitly so that the tested biases are accessible uniformly and consistently to all workers. We did this selectively to certain groups by stating that "From our previous data, there are on average 60 devices visible in each image." In this way, we could create the situation of workers being primed with a bias that is very different from the true answer. We note this form of priming is more direct than in other studies (such as (Horton et al., 2010)) but quite representative of real situations, where workers have easy access to common information that allows to guess a heuristic answer.

We tested that the priming did indeed introduce the bias expected through the anchoring effect by comparing the answer distributions with and without priming. These are treatments 1-3 in Table 1, explained further below. While the average of answers without priming was quite close to the true value with a mean error of 1.0667, with priming to an incorrect value the mean error jumped to 5.6316. When priming to the correct value, the error also increases somewhat, to 2.9434. This is explained by the fact that anchoring perturbs answers from purely random errors. Similar to the anchoring effect as originally reported in (Tversky and Kahneman, 1974), the answer distributions shown in Figure 3 are shifted from those in Figure 2 towards the primed value without showing a peak at the primed value itself.

Explaining the game-theoretic incentive schemes to workers is a big challenge, and misunderstandings will perturb the experimental results. We therefore translated the game-theoretic bonus computation into a set of approximate payments that reflect the same incentives. We used the following payment and bonus schemes, which all work out to an expected payment of $0.05:

1. *no bonus:* payment = $0.05
2. *vague:* "We will pay you $0.01, and a bonus of $0.04 if you provide a reasonable count."
3. *peer confirmation:* "We will pay you $0.01, and a bonus of $0.04 if your response matches that of another worker on the same image within +/- 1 tolerance."
4. *Peer Truth Serum:* "We will pay you $0.01, and a bonus if your response matches that of another worker on the same image within +/- 1 tolerance. The bonus is as follows:
   - $0.01 if the matching answer is within +/-1 of the current average count.
   - $0.06 if the matching answer is something different."

   Implementing the Peer Truth Serum by applying the formula would be hard to understand and lead to payments that are not expressible as whole cents. Therefore, the payments were derived from a game-theoretic analysis by dividing the answer spectrum into 5 intervals, of which one is the interval around the primed value. It corresponds to an assumption that the probability the value falling close to the primed value is 0.6, and that of falling in one of the other intervals is 0.1. The expected bonus under those assumptions is $0.0364 and thus quite close to that of peer confirmation; also the expected bonus is $0.04 which is exactly equal to that of peer confirmation. Therefore, we believe this bonus might be considered equal by the workers.
5. Bayesian Truth Serum: using a simplified version of the BTS formula to compute the bonus and the same text used in the study in (Shaw et al., 2010).

The Peer Truth Serum is always designed for a certain expected bias, and so the user study controls this bias uni-

Run	Bonus	Priming	Mean Error
1	no bonus	none	1.0667
2	no bonus	60	5.6316
3	no bonus	34	2.9434
4	vague	none	2.2500
5	vague	60	6.6563
6	vague	34	9.0984
7	peer confirmation	none	0.3492
8	peer confirmation	60	3.3429
9	peer confirmation	34	2.4194
10	Peer Truth Serum	60	0.8000
11	Peer Truth Serum	34	2.1667
12	Bayesian Truth Serum	none	7.2323
13	Bayesian Truth Serum	60	11.7439
14	Bayesian Truth Serum	34	8.1616

Table 1: *The 14 different treatments that were evaluated.*

formly for all users by explicit priming. We evaluated all treatments with priming to 34 and 60 objects, and all except PTS without any priming. Thus, we ran the experiment with 14 different treatments, shown in Table 1, on the Amazon Mechanical Turk platform.

In launching the tasks, it is important to ensure that workers cannot self-select among different treatments, and that they do not suspect that they are tested on different treatments. We faced the choice on whether to attribute different treatments to workers randomly or in a round-robin fashion, or whether to launch the tasks in batches of identical treatments. Assigning the tasks in a round-robin fashion has the advantage of being robust to variations in the worker population over time, but on the other hand would make the nature of the experiment obvious to anyone who revisited the task. Launching the treatments in batches ensures that the nature of the experiment is hidden, but is susceptible to variations in worker population. We guarded against variations in the population by launching each treatments in two batches of 50 tasks each, one in the morning and one in the evening, and only during weekdays.

With this design, at any given time only a single treatment was open to ensure that workers could not self-select between different treatments and thus bias the results. To ensure that tasks in the different treatments were executed by different workers, we discarded all data from workers that had already solved an earlier task using the same image. Even for the last treatments we still had a large portion (over 50%) of workers that had never participated in any version of the task before.

Tasks were run at 2 different times of the day to correct for bias due to worker origin. We queried the demographics and satisfaction of workers using two questionnaires after completing the task. There is an even balance of male and female workers, a relatively high education level and a good mix of professions. Compared to the Mechanical Turk surveys in (Ipeirotis, 2010), there were more Asian and more younger workers but the demographics agree in gender distribution and education level.

To ensure that tasks were not misunderstood or upset

Bonus	Priming	Average Error	t-test
no bonus	none	1.0667	
	60	5.6316	p = 0.0266
	34	2.9434	p = 0.2092
vague	none	2.2500	
	60	6.6563	p = 0.0810
	34	9.0984	p = 0.0032
peer conf.	none	0.3492	
	60	3.3429	p = 0.0554
	34	2.4194	p = 0.1496
peer conf. PTS	none	0.3492	
	60	0.8000	p = 0.4036
	34	2.1667	p = 0.2145

Table 2: *Comparison of error between treatments without and with priming to an incorrect value (60) and a correct value (34). The Peer Truth Serum only applies in case of priming so it is compared against the Peer Confirmation that pays the same everywhere. p-values are with respect to the baseline shown above each treatment.*

workers, we also asked several questions about worker satisfaction. For Treatments 1 through 11, we find that 98% would do the task again, 95% considered the instructions clear, and only 9% of the workers found the task difficult, whereas 29% found it easy and 60% average. Treatments 12 through 14 gave very different results, discussed below.

Since counting errors are likely to follow a Gaussian distribution, the mean is the best estimator. We thus compared the difference between the mean of the values reported by the workers and the true value.

Results

Effect of priming without bonus

The effect of priming shows up clearly in the distributions of answers. Figures 3 and 2 illustrate that the true value is much harder to see in the distribution obtained with priming than in the one without priming, and all techniques of averaging (mean/median/mode) that would give an almost perfect result without priming would produce a significant error on this distribution. The quantitative effect of priming on the mean reports can be seen in Table 2. Without bonus (first row of the table), priming to an incorrect value results in a strong increase of the average error, and indeed the difference in answer distributions is significant with a p-value of 0.0266. When bonus schemes are applied, priming also consistently degrades accuracy, although the difference is slightly above the p-value that is considered significant (p=0.081 and p=0.0554). At least for the case where no bonus scheme is used, we can clearly conclude:

Result 1: Priming with an incorrect value decreases accuracy while degradation is not significant for priming with an accurate value.

Peer Truth Serum and priming

Figure 4 shows the error distribution of answers when the Peer Truth Serum is applied, and it can clearly be seen to en-

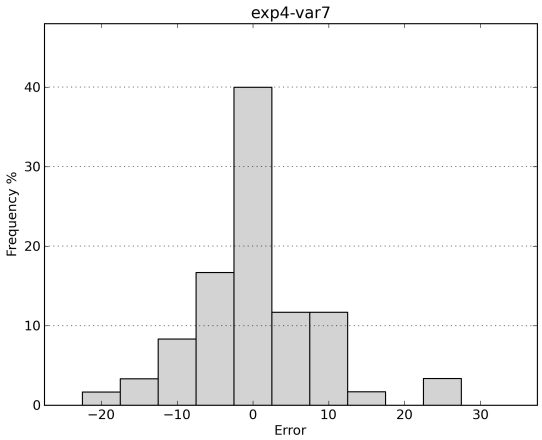

Figure 4: *Distribution of answers with priming when applying the Peer Truth Serum.*

courage a more balanced distribution and a clear peak at the true value. More precise results can be seen in Table 3. Without bonus, priming to an incorrect value causes a larger error of 5.6316, while using the bonus scheme this is reduced by 85% to just 0.8000, with a strong significance indicated by the p-value of 0.0088. When priming is to the correct value, the Peer Truth Serum seems to only have a minor positive effect: we have an average error of 2.1667 vs. 2.9434 and the t-test gives a p-value of 0.3731, indicating no significant difference. Therefore we conclude:

Result 2: The Peer Truth Serum counters priming better than having no bonus.

A possible alternative explanation is that workers select tasks according to the advertised payment, which was higher ($0.05) for the treatment without bonus than for the treatments with bonus. To consider this aspect, we also ran the treatment without bonus and payment of $0.01. It resulted in lower average error for no priming (0.1493) and priming to 60 (2.8438) but higher for priming to 34 (4.2609). It would thus make the bonus schemes better also for the case of correct priming, but besides lower p-values it does not qualitatively change the results elsewhere. We do not consider this a plausible explanation of the observed behavior.

Table 3 shows the performance of different bonus schemes in countering the negative effects of priming. The vague scheme always performs poorest, in fact worse than having no bonus at all. Peer confirmation gives some improvement, although it is not statistically significant. The Peer Truth Serum gives the best results for both priming to an incorrect and a correct value, and it is the only scheme that provides a statistically significant improvement for priming to an incorrect value. When priming to a correct value, there is a slight but statistically insignificant improvement.

We also compared PTS to the closest contender, peer confirmation. When priming with 60, PTS is better than peer confirmation with an almost significant p=0.0618. When priming with 34, they are not significantly different with p=0.455.

Bonus Scheme	Priming	Average Error	t-test
none	60	5.6316	
vague	60	6.6563	p = 0.3782
peer conf.	60	3.3429	p = 0.1306
PTS	60	0.8000	p = 0.0088
none	34	2.9434	
vague	34	9.0984	p = 0.0110
peer conf.	34	2.4194	p = 0.4020
PTS	34	2.1667	p = 0.3731

Table 3: *Comparison of error for different bonus schemes when priming is present.*

Thus, we conclude:

Result 3: The Peer Truth Serum corrects priming better than other bonus schemes.

Bayesian Truth Serum

The three treatments we ran using the Bayesian Truth Serum all produced results with very bad accuracy ranging from an error of 7.23 with no bias to 11.74 when primed with 60. The post-study surveys showed that the scheme is complex and confuses at least some fraction of the workers. The percentage of workers that found the task difficult rises from 9% to 28.5%, the percentage who would do the task again drops from 98% to 84%, and the percentage who thought that the instructions were clear dropped from 95% to 78%.

Conclusions

Heuristic biases arise when workers have expectations of answers that make them adopt biased heuristic strategies rather than actually solve the task. Bias is a big problem for crowdsourcing as errors cannot be eliminated by the usual strategy of increasing the number of workers.

We have constructed a task that easily allows to set a specific and uniform bias by the experimenter, and shown that this bias indeed significantly reduces accuracy.

Our work confirms the observations of (Shaw et al., 2010; Huang and Fu, 2013) that bonus schemes that motivate workers to match other workers' answers have the best chance of success. The peer confirmation scheme, which pays a fixed reward when the answer closely matches that of another worker, provides an accuracy improvement when bias is not present. However, in the presence of bias, it turns out to be important to also scale the payments. We have investigated the Peer Truth Serum as an improvement of Peer Consistency that implements this and it shows great performance at eliminating bias, improving accuracy by 85% and even improving over the accuracy in the unbiased case.

Due to the constraints of a live user study, we were not able to test the game-theoretic scheme exactly, but an approximation. The lesson for task designers is that bonuses in peer consensus should be scaled roughly inversely proportional to the likelihood of the answer. Given the uncertainties of worker beliefs on online labor markets, we do believe it

makes sense to implement the game-theoretic scheme more precisely.

We believe that while the accuracy improvements that can be obtained with bonus schemes can be achieved equally well by increasing the number of workers, bonus schemes are the only way to eliminate bias. While the Peer Truth Serum is a step in this direction, it requires knowing about the likely bias. One way around this would be to observe the bias from the distribution of answers to similar tasks, and in a forthcoming paper we describe the game-theoretic principles of such a scheme (Radanovic and Faltings, 2014b). Another possibility are approaches such as the Bayesian Truth Serum, where workers themselves report their bias. More work is required to simplify such schemes so that they can be used with human computation platforms.

References

L. von Ahn and L. Dabbish. Labeling Images with a Computer Game. *ACM Conference on Human Factors in Computing Systems*, pp. 319-326, 2004.

A. Dasgupta and A. Ghosh: Crowdsourced Judgement Elicitation with Endogenous Proficiency. WWW Conference, 2013

B. Faltings, J.J. Li and R. Jurca. Incentive Mechanisms for Community Sensing. *IEEE Transactions on Computers*, **63**(1), pp. 115-128, 2014.

X. A. Gao, A. Mao, Y. Chen and R. P. Adam. Trick or Treat: Putting Peer Prediction to the Test. Proc. of the 15th ACM Conference on Economics and Computation (EC), 2014.

F. Garcin and B. Faltings. Swissnoise: Online Polls with Game-Theoretic Incentives. Proceedings of the 26th IAAI, 2014.

C. G. Harris. You're Hired! An Examination of Crowdsourcing Incentive Models in Human Resource Tasks. WSDM 2011 Workshop on Crowdsourcing for Search and Data Mining, pp. 15-18, 2011.

J. J. Horton, D.G. Rand and R. Zeckhauser. The Online Laboratory: Conducting Experiments in a Real Labor Market. Faculty research paper RWP10-017, Harvard Kennedy School, 2010.

S.-W. Huang and W.-T. Fu. Enhancing Reliability Using Peer Consistency Evaluation in Human Computation. *Proceedings of the 2013 conference on Computer supported cooperative work*, pp. 639-648, 2013

S.-W. Huang and W.-T. Fu. Don't Hide in the Crowd! Increasing Social Transparency Between Peer Workers Improves Crowdsourcing Outcomes. *CHI*, pp. 621-630, 2013

P. Ipeirotis. Demographics of Mechanical Turk. New York University Working Paper No. CEDER-10-01, 2010.

P. Ipeirotis, F.Provost and J. Wang. Quality Management on amazon Mechanical Turk.*ACM SIGKDD Workshop HCOMP'10*, pp. 64-67, 2010.

J. Lorenz, H. Rauhut, F. Schweitzer and D. Helbing. How social influence can undermine the wisdom of crowd effect. Proceedings of the National Academy of Sciences, 108, pp. 9020-9025, 2011.

R. Jurca and B. Faltings. An Incentive Compatible Reputation Mechanism. Proceedings of the IEEE Conference on E-Commerce, pp. 285-292, 2003.

R. Jurca and B. Faltings. Mechanisms for Making Crowds Truthful. *Journal of Artificial Intelligence Research (JAIR)*, 34, 2009, pp. 209-253.

R. Jurca and B. Faltings. Incentives for Answering Hypothetical Questions. *Workshop on Social Computing and User Generated Content*, ACM Conference on Electronic Commerce, San Jose, 2011.

E. Kamar and E. Horvitz. Incentives for Truthful Reporting in Crowdsourcing. *Proceedings of AAMAS 2012*, Valencia, 2012

N. Lambert and Y. Shoham. Eliciting Truthful Answers to Multiple-Choice Questions. In *Proceedings of the tenth ACM conference on Electronic Commerce*, pp. 109-118, 2009.

N. Miller, P. Resnick, and R. Zeckhauser. Eliciting Informative Feedback: The Peer-Prediction Method. *Management Science*, 51:1359–1373, 2005.

A. Papakonstantinou, A. Rogers, E. H. Gerding, and N. R. Jennings. Mechanism design for the truthful elicitation of costly probabilistic estimates in distributed information systems. *Artif. Intell.*, 175(2):648–672, 2011.

D. Prelec. A Bayesian Truth Serum for Subjective Data. *Science*, 306(5695), pp. 462-466, 2004.

G. Radanovic and B. Faltings. A Robust Bayesian Truth Serum for Non-binary Signals. *Proceedings of the 27th AAAI Conference on Artificial Intelligence (AAAI'13)*, pages 833-839, 2013

G. Radanovic and B. Faltings. Incentives for Truthful Information Elicitation of Continuous Signals. *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI'14)*, 2014

G. Radanovic and B. Faltings. Incentives for Effort in Crowdsourcing using the Peer Truth Serum. *submitted for publication*, 2014.

L. J. Savage. Elicitation of Personal Probabilities and Expectations. *Journal of the American Statistical Association*, 66(336):783–801, 1971.

A. Shaw, J. J. Horton and D. L. Chen. Designing Incentives for Inexpert Human Raters. Proceedings of the ACM Conference on Computer Supported Cooperative Work, 2010

A. Tversky and D. Kahneman. Judgment under uncertainty: Heuristics and biases. *Science*, 185, pp. 11241130, 1974

J. Witkowski and D. C. Parkes. Peer Prediction without a Common Prior. *Proceedings of the 13th ACM Conference on Electronic Commerce (EC 2012)*, 2012.

J. Witkowski and D. C. Parkes. A Robust Bayesian Truth Serum for Small Populations. *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI 2012)*, 2012.

A. Zohar and J. S. Rosenschein. Robust Mechanisms for Information Elicitation, in AAAI '06: Proceedings of The Twenty-First National Conference on Artificial Intelligence, July 16-20, AAAI Press, Menlo Park, CA, USA, 2006.