Deep Bayesian Trust : A Dominant Strategy and Fair Reward Mechanism for Crowdsourcing

Naman Goel*, Boi Faltings †
Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland

April 17, 2018

Abstract A common mechanism to assess trust in crowdworkers is to have them answer *gold* tasks. However, assigning gold tasks to all workers reduces the efficiency of the platform. We propose a mechanism that exploits transitivity so that a worker can be certified as trusted by other trusted workers who solve common tasks. Thus, trust can be derived from a smaller number of gold tasks assignment through multiple layers of peer relationship among the workers, a model we call *deep* trust. We use the derived trust to incentivize workers for high quality work and show that the resulting mechanism is dominant strategy incentive compatible. We also show that the mechanism satisfies a notion of fairness in that the trust assessment (and thus the reward) of a worker in the limit is independent of the quality of other workers.

1 Introduction

As supervised machine learning algorithms require large amounts of labeled data, crowdsourcing has emerged as a preferred method to conveniently obtain such data within reasonable budget constraints. Requesters publish their unlabeled data on crowdsourcing platforms, multiple workers look at the data instances (HITs) and report their labels for a small amount of money. Unfortunately, in this method there are no guarantees about the quality of labels. The quality can get significantly degraded if tasks require some effort. Hence, there is a need to align the incentives of crowdworkers and encourage them to provide quality data. This can be achieved by rewarding the workers with a performance-based bonus in addition to the promised fixed payment.

One common way to reward the workers is to use gold standard tasks. The gold standard tasks are tasks for which the correct answers are already known to the requester. These tasks don't bring any additional value to the requester and their sole purpose in crowdsourcing is quality control. Consider the common technique, in which the requester inserts a set of gold standard tasks in the batch of tasks to be solved by every worker. In this technique, the payments to the workers are made contingent on their performance on the gold tasks. Using the common assumption that workers solve all tasks with same proficiency, the quality of workers' answers can be judged based on their performance on the gold tasks. Rewards based on this estimated quality incentivizes every worker for performing high quality work as a dominant strategy (i.e. independent of the quality of other workers). However, it is not difficult to see the weakness of this technique. Since every worker solves gold tasks and the requester already knows their correct answers, it leads to a waste of task budget of the requester. Moreover, to reduce the variance in the payments, one may require a sufficient number of gold tasks to be solved by every worker. This further amplifies the problem of task budget waste. Also, note that the technique only works if the identity of gold tasks in not leaked. If the requester uses the same gold set

*naman.goel@epfl.ch
†boi.faltings@epfl.ch

repeatedly, there is a risk of the gold tasks being leaked [22]. On the other hand, using a different gold set every time is not scalable.

In this paper, we introduce a novel reward mechanism that dominant strategy incentive compatibility while assigning gold tasks only to a small (constant) number of workers. We call our mechanism the ***Deep Bayesian Trust* Mechanism**. The mechanism works for tasks that are objective in nature (i.e. for which there is an underlying ground truth) and have discrete valued answers. We use the Dawid-Skene model for workers' ability, which doesn't require the assumption that honest workers are able to always give correct answers. Since our mechanism ensures dominant strategy incentive compatibility instead of only Nash equilibria, it takes care of the real world situation in which some workers may inevitably deviate to different strategies.

Our mechanism is based on the observation that in large scale crowdsourcing settings, every worker reports answers to many similar tasks and hence the joint distribution of their answers can be very useful in learning something about one worker given some information about another worker. It starts by rewarding a fixed set of workers based on gold tasks and then uses the answers provided by the workers on non-gold tasks as *contributed* gold tasks to reward more workers. It continues this *deep* chain of arbitrary depth and exponentially increasing number of workers until all the tasks have been solved by the required number of workers. Since the answers of any two workers solving common tasks must be correlated, trusted information about the correctness of one worker also gives information about the correctness of other workers. As we show in the paper, this can be done using the empirical estimate of the joint distribution of their answers on common tasks and Bayes' rule.

As fairness of algorithms affecting humans is becoming a critical issue, it is important to justify the fairness of algorithms that determine payments of workers. We, perhaps for the first time, address the issue of fairness of crowdsourcing reward mechanisms in a principled way and show that our mechanism ensures fair rewards to workers. The summary of our main contributions is as follows :

- We propose a dominant strategy incentive compatible (DSIC) mechanism, called the ***Deep Bayesian Trust* Mechanism**, which rewards a limited number of workers with gold tasks and the rest using peers' answers.
- We define the notion of fairness in rewards for crowdworkers and show that our mechanism ensures fair rewards to every worker.
- The DSIC and fairness result is stronger than the recently proposed unique Nash equilibrium result of [5] in similar settings and doesn't require any information about peers to be communicated to the workers.
- We show, through numerical simulations, the robustness of our mechanism to various reporting strategies of the workers in different realistic scenarios. We also discuss results of a preliminary study conducted on Amazon Mechanical Turk to support the use of our mechanism in improving the quality of responses.

2 Related Work

Quality control in crowdsourcing [12] is a well recognized research issue. On one hand, there are post processing approaches [14, 27, 17] to clean/aggregate the data collected from different workers and on the other hand, there are efforts to collect as high quality data as possible through better tasks design [15], assignment [19, 33] and incentives schemes [10]. There are also approaches to make machine learning algorithms robust to quality issues in data collected from crowdsourcing [21]. The approaches are complementary in nature. The root cause of the quality issues is the strategic nature of workers [6] and their incentives [32, 16] not being properly aligned. This paper addresses the problem of aligning the incentives of the workers.

The research in incentive schemes for crowdsourcing is divided into two categories. The first category of work assumes no gold standard tasks to be available. Such schemes are called the peer-consistency schemes. The seminal work in this category is due to Dasgupta and Ghosh [3]. It models the settings where workers' invested effort determines their proficiency on binary evaluation tasks. The main idea in this work is to reward the workers for agreeing with one another on a given common task but also penalize them for blindly agreement. The penalty term is learned from the statistics of the workers' reports on other non-common tasks solved by them. The mechanism makes the truthful equilibrium the most profitable equilibrium (except the permutation equilibria). The Peer Truth Serum [26] generalizes this to non-binary settings and subjective tasks. The idea is to use the distribution of answers reported for statistically similar tasks as the prior probability of answers for the task. A worker gets a reward that is inversely proportional to the prior probability of his answer, but only if the answer matches the answer of a peer on the particular task. This mechanism shows similar incentive properties under the *self-prediction* assumption on workers' beliefs. Other results in this category of incentives scheme include [20, 24, 29, 2, 25, 30, 31, 29, 2, 18, 11]. All the peer prediction mechanisms suffer from the problem of undesired equilibria.

The second category of work assumes there are some gold standard tasks available. The two works that are most closely related to ours are the following. The mechanism of [8] randomly selects a few workers and spot checks only those workers with an oracle. The rest of the workers are given a constant amount of reward. While it is theoretically sufficient, a low probability increases the volatility of payments. The payments of the workers with gold sets need to be scaled appropriately to ensure dominant strategy incentive compatibility. Attempt to reduce this volatility by scaling up the constant payment will result in overall inflation of payments. Another undesired implication of low probability spot checking is that it may take longer for workers to learn about the scheme. Finally, because of low probability spot checking, the task requester would hardly get any idea of workers' reliability. The mechanism of [5] arranges workers in a hierarchy. The workers at the top of the hierarchy are evaluated by a trusted supervisor or oracle. The workers below that level are evaluated by the workers above them. However, this scheme offers comparatively weaker incentive compatibility (unique Nash equilibrium). For the result to hold, it is also required that workers are aware of their level in the hierarchy.

Finally, there is also some interesting experimental work [9, 10] on evaluating the effect of incentive mechanism in crowdsourcing.

3 Model

We consider large scale crowdsourcing settings where multiple workers provide answers of many micro-tasks requiring human intelligence. The tasks are assumed to be of objective nature (i.e. they have a verifiable unknown ground truth answer). The tasks have a discrete answer space $\{1, 2, ..., K\}$ of size K. Examples of such tasks include identifying the presence of a certain object in a given image, measuring the level of pollution at a given place according to a certain standard, verifying the correctness of some objective information available on the web etc. For any task, our model has three random variables. The first is the unknown **ground truth** G answer for the task. The second is the worker i's **observed answer** X_i that she obtains on solving the task. X_i is worker's private information. The third is the worker's **answer** Y_i that she actually reports (reveals) to the requester as her answer for the task. We use lower case letters $g, x_i, y_i \in \{1, 2, ..., K\}$ to denote realizations of these random variables and will drop the subscript i for brevity, where it is not required.

Definition 1 (Proficiency Matrix)**.** *The proficiency matrix A_i for a worker i is a $K \times K$ right stochastic matrix with element $A_i[g, x]$, for $g, x \in \{1, 2, ..., K\}$, being the probability of the worker obtaining answer to a task as x given that the ground truth is g.*

The definition of proficiency matrix is motivated by the Dawid-Skene model [4], which is a widely accepted model in literature. The proficiency matrix models the ability of a worker to obtain correct answers in a very flexible way. Every worker can have a different proficiency matrix that models bias towards each answer. The elements in every row sum to 1 and so it is right stochastic.

Definition 2 (Strategy Matrix)**.** *The strategy matrix S_i for a worker i is a $K \times K$ right stochastic matrix with element $S_i[x, y]$, for $x, y \in \{1, 2, ..., K\}$, being the probability of the worker's reported answer on a task being y given that her observed answer is x.*

The strategy matrix models every possible strategy that a worker may play in revealing her private information (obtained answer x). Two of the most common strategies are the honest (or truthful) and heuristic strategies as defined below. Mixed strategies, for example being truthful and heuristic with certain probabilities, can also be modeled with this strategy matrix using a convex combination of the two strategy matrices.

Definition 3 (Truthful Strategy)**.** *A strategy is called truthful if the strategy matrix is an identity matrix.*

In a truthful strategy, a worker solves the tasks and reports her answers as obtained.

Definition 4 (Heuristic Strategy)**.** *A strategy is called heuristic if all rows of the strategy matrix are identical.*

In a heuristic strategy, a worker reports independently of the obtained answer. *That also includes the strategy in which workers don't obtain any answer and report randomly*.

In crowdsourcing, the level of effort to be invested is also a strategic decision for the workers. Our definition of the strategy matrix may seem to not model effort explicitly. As we will discuss later, this only simplifies the analysis without restricting the applicability of our results when effort is a continuous variable affecting workers' proficiency and eliciting maximum effort is the desired outcome.

Definition 5 (Trustworthiness Matrix)**.** *The trustworthiness matrix T_i for a worker i is a $K \times K$ right stochastic matrix with element $T_i[g, y]$, for $g, y \in \{1, 2, ..., K\}$, being the probability of the worker's report on a task being y given that the ground truth is g.*

Note the difference between the proficiency and trustworthiness matrices. Proficiency models worker's ability while trustworthiness is a function of her ability and honesty.

Proposition 1. *The trustworthiness matrix T_i of a worker i is given by*

$$T_i = A_i S_i$$

where A_i is the proficiency matrix and S_i is the strategy matrix of worker i.

The proof is trivial by using Bayes' rule since worker's report and ground truth are conditionally independent, given the answer she obtained.

Definition 6 (Oracle)**.** *An oracle is any trusted agent o, whose proficiency matrix is known and whose strategy matrix is an identity matrix.*

The oracle can be an expert, who is always known to report the answers as observed and whose proficiency is known. If the oracle is the source of gold standard answers, then by definition, its proficiency matrix can also be assumed to be an identity matrix.

Definition 7 (Dominant Strategy Incentive Compatibility)**.** *A reward mechanism is called dominant strategy incentive compatible if the expected reward of any worker is strictly maximized by playing a truthful strategy, no matter what strategies other workers use.*

It is a very strong form of incentive compatibility that we desire to achieve. The reward is *strictly* maximized by an honest strategy even if others are not honest. The truthful strategy thus dominates, among others, any heuristic strategy of not solving the tasks and reporting randomly. It also dominates any mixed strategies.

4 Preliminaries

As explained in the model, trustworthiness is a function of a worker's ability and strategy (both being unknown). In this section, we explain the main building block of our mechanism : the process of estimating the trustworthiness of a worker, given the trustworthiness of another worker by using the joint distribution of their reported answers on shared tasks.

Definition 8 (Peer)**.** *For every worker i, the mechanism assigns another worker j as her peer. Workers i and j are assigned sets of tasks Q^i and Q^j respectively, such that $|Q^i \cap Q^j| \gg 0$.*

The definition requires that some tasks are solved by both the worker and her peer. Both workers also solve some other tasks that are not overlapping.

N^{ij}	Number of shared tasks between worker and peer ($|Q^i \cap Q^j|$)
$N^{ij}(condition)$	Number of shared tasks between worker and peer ($|Q^i \cap Q^j|$) for which $condition$ is true. For example $N^{ij}(Y_i = 1, Y_j = 2)$ is the number of shared tasks on which worker i reported 1 and worker j reported 2.
$\omega(E1)$	Empirical relative frequency of an event $E1$ in the sample of reports provided by workers i and j on shared tasks. For example $\omega(Y_i = 1)$ is the relative frequency of the worker i's report being 1 on the shared tasks, i.e. $\omega(Y_i = 1) = \frac{N^{ij}(Y_i=1)}{N^{ij}}$
$\omega(E1/E2)$	Empirical conditional relative frequency of event $E1$ given event $E2$. For example $\omega(Y_i = 1\|Y_j = 2)$ is the conditional relative frequency of the worker i's report being 1 on the shared tasks for which worker j's report is 2, i.e. $\omega(Y_i = 1\|Y_j = 2) = \frac{N^{ij}(Y_i=1,Y_j=2)}{N^{ij}(Y_j=2)}$
$P(G = g)$ or $P(g)$	The prior probability of the ground truth answer of any randomly selected task being g. Assumed to be known and fully mixed ($P(g) > 0 \; \forall g \in \{1, 2, ..., K\}$).

Table 1: Notations

Let T_j be the known trustworthiness matrix of worker j and let j be the peer of another worker i, whose trustworthiness matrix T_i is not known. Please refer to the notation Table 1 for the following lemma.

Lemma 1. *As $N^{ij} \to \infty$, the following holds w.h.p.*

$$\omega(Y_i = y_i|Y_j = y_j) = \sum_{g \in \{1,2,\ldots,K\}} T_i[g, y_i] \cdot \Big(\frac{T_j[g, y_j] \cdot P(g)}{\omega(Y_j = y_j)}\Big) \tag{1}$$

$\forall \; y_i, y_j \in \{1, 2, ..., K\}$ *and* $\omega(Y_j = y_j) \neq 0$

Proof (Sketch). The main idea in the proof is that, in the limit, the relative frequency terms $\omega(\cdot|\cdot)$ can be substituted by the corresponding probability terms $P(\cdot|\cdot)$ by the law of large numbers and continuous mapping theorem. The expressions on the right side of the Equations follow from the application of Bayes' rule for conditional probability since the reports of workers i, j on shared tasks are conditionally independent given the ground truth. □

Remark : The proof assumes that the shared tasks are of similar difficulty so that a given worker's proficiency doesn't vary much with tasks in a given batch. This assumption is common in most game theoretic mechanisms as well as in answer aggregation algorithms for crowdsourcing [4]. We assume that workers are strategically indifferent towards different tasks that are assigned to them since the tasks are indistinguishable. The assumption about task indistinguishability is crucial in all mechanisms involving gold tasks. Note that the assumption does **not** rule out the possibility of workers playing mixed strategies. This is because any mixed strategy, can be modeled by the strategy matrix defined earlier (as a convex combination of *pure* strategies).

The linear system of Equations in 1 can be solved for the only unknowns $T_i[g, y_i] \; \forall \; g, y_i \in \{1, 2, ..., K\}$ irrespectively of the strategies played by the workers, provided the system is *well-defined*. It requires that $\omega(Y_j = y_j) \neq 0$ and for a unique solution, the coefficient matrix of this linear system should have linearly independent rows. More precisely, it requires that the posterior distributions $P(G = g|Y_j = y_j)$ over $g \in \{1, 2, ..., K\}$ for any two different $y_j \in \{1, 2, ..., K\}$ are not identical. One interesting example, where such situation may arise, is when the peer j plays a heuristic strategy. In such a case, the reports are not correlated with the ground truth and the posterior distributions $P(g|Y_j = y_j)$ are same as the prior distribution $P(g)$.

As an example, consider the binary answer space ($K = 2$). If $T_j[1, 1] + T_j[2, 2] \neq 1$, solving the system of Equations in 1 gives the following,

$$T_i[1,1] = \frac{b \cdot \omega(y_i = 1|y_j = 1) - (1 - a) \cdot \omega(y_i = 1|y_j = 2)}{ab - (1 - a)(1 - b)}$$

$$T_i[2,2] = 1 - \frac{a \cdot \omega(y_i = 1|y_j = 2) - (1 - b) \cdot \omega(y_i = 1|y_j = 1)}{ab - (1 - a)(1 - b)}$$

where, $a = \frac{T_j[1,1] \cdot P(1)}{\omega(y_j = 1)}$ and $b = \frac{T_j[2,2] \cdot P(2)}{\omega(y_j = 2)}$.

$T_i[1, 1]$ and $T_i[2, 2]$ together define the trustworthiness matrix of i in binary answer space. For non-binary case also, many libraries are available to solve system of linear equations efficiently. It may be noted that the requirement for this estimation method to work in binary answer spaces ($K = 2$) is weaker than the general case ($K > 2$). It only requires that the reports of the peer j are not independent of the ground truth ($T_j[1, 1] + T_j[2, 2] \neq 1 \implies T_j[1, 1] \neq T_j[2, 1] \implies T_j[2, 2] \neq T_j[1, 2]$). As long as the reports of peer j have some correlation with the ground truth, we can estimate the trustworthiness matrix of worker i in binary answer space. For the general case, the requirement is as discussed earlier.

We now use this method to develop our mechanism which transitively estimates the trustworthiness of workers and rewards them using their trustworthiness scores.

5 The Deep Bayesian Trust Mechanism

Our mechanism follows large scale crowdsourcing settings in which a requester publishes several batches of tasks and workers self-select to solve the tasks. The mechanism (shown in Mechanism 1) maintains a pool of workers' answers which are suitable for evaluating other workers. The pool is initialized with some tasks and their answers given by the oracle. The trustworthiness matrix of the oracle is initialized to be T_o. The mechanism then publishes several batches of tasks on the platform such that each batch has some tasks in common with tasks solved by the oracle and some unique new tasks. Workers self-select themselves to solve one batch each and report their answers for respective batches. In other words, the oracle becomes a peer of each of these workers. Let's assume that the oracle solves s_o tasks. The mechanism publishes k batches of tasks such that there are s_o tasks in common with the oracle and s_n unique new tasks in each batch. Thus, it publishes $k \cdot s_n$ tasks that are new (not solved by oracle already) and also $k \cdot s_o$ instances of the same s_o tasks that are already solved by the oracle. k becomes a hyper-parameter of the mechanism. These are the first 3 steps of the mechanism as listed in Mechanism 1. As the workers submit their respective batches, the mechanism also rewards the workers asynchronously for their answers using a reward function as described in step 4 of the mechanism. To calculate the reward, the mechanism uses Lemma 1 to estimate the trustworthiness matrix of the answers given by workers. The reward for worker i is given by $R_i = \Big(\sum_{k \in \{1,2,...,K\}} T_i[k,k] \Big) - 1$. The worker gets her reward and is now out of the mechanism. At this stage, the mechanism takes a decision about reusing the answers provided by the workers. If the worker's answers satisfy a certain criterion, they are added (grouped by the worker id) to the pool. For now, we consider this filtering criterion as a black box that ensures only suitable workers' answers are added to the answer pool. If the worker's answers are added to the pool, the mechanism immediately publishes more batches such that there are some tasks in common with the new (non-gold) tasks just solved by the previous worker and some more new tasks in each of the batches. This step is the same as described earlier, the only difference is that now the batches being published have tasks in common with the tasks solved by a worker who is not the oracle. Thus, the peers are now fellow workers not the oracle. These steps are repeated in parallel and asynchronously until the mechanism has obtained the desired number of answers for all its unsolved tasks. These constitute steps 5 and 6 of Mechanism 1.

To summarize, the mechanism starts with an answer pool seeded with the oracle's answers, uses the answers in the pool to assess trust in other workers, expands this pool based on the trustworthiness of the workers and repeats the process. The main result of our paper (Theorem 1) is that irrespectively of which batch a worker gets to solve or who acts as her peer, it is her dominant strategy to be truthful.

It may be noted that the mechanism doesn't assign any permanent reputation to workers. It just evaluates the answers provided by a worker in any given batch and adds them to an answer pool together with an estimate of the trustworthiness in those answers.

Mechanism 1 : The Deep Bayesian Trust Mechanism

---

1. Assign a set of tasks to the oracle o and obtain its answers on the tasks.

2. Initialize an *answer pool* AP with the answers given by oracle.

   $AP = \Big[\big[o : T_o : q_1 - \mathbf{a_1}, q_2 - \mathbf{a_2}, q_3 - \mathbf{a_3}, ... \big] \Big]$
   o stands for oracle, T_o is the trustworthiness of the oracle and $q_i - a_i$ are the task-answer pairs provided by the oracle.

3. Publish batches of tasks such that each batch has some tasks in common with one of the randomly chosen peer who has contributed answers in the answer pool AP. Rest of the tasks in the batches are new.

4. For any worker i who submits her batch, solve the system of Equations in 1 to find unknowns (i.e. trustworthiness T_i). Reward worker i for her answers with $R_i = \Big(\sum_{k \in \{1,2,...,K\}} T_i[k,k] \Big) - 1$.

5. If the answers of worker i satisfy the pool expansion criterion, add the answers to the pool and assign them trustworthiness T_i as obtained in Step 4.

   For example, at a given instant, the pool may look as follows :
   $AP = \Big[\big[o : T_o : q_1 - \mathbf{a_1}, q_2 - \mathbf{a_2}, ... \big], \big[W_1 : T_{W_1} : q_2 - \mathbf{a_2}, q_4 - \mathbf{a_4}, ... \big], \big[W_2 : T_{W_2} : q_2 - \mathbf{a_2}, q_5 - \mathbf{a_5}, ... \big], ... \Big]$
   Here W_1, W_2.... are the identities of workers, followed by their trustworthiness and their task-answer pairs.

6. Repeat steps 3, 4 and 5 for every submitted batch *asynchronously*, until desired number of answers are collected for all tasks.

Pool Expansion Criterion

We now discuss the pool expansion criterion which was omitted while discussing the idea of the mechanism. The main idea is to check that if the answers provided by worker i can be used to estimate the trustworthiness of another worker or not. As discussed in Section 4, this depends on whether the coefficient matrix of the linear system of equations has linearly independent rows or not. The interesting observation here is that this depends only on the answers of this worker and not on another worker who may get this worker as her peer. For example, assume that the answers of worker i are added to the pool and let f be another worker who gets i as her peer. In that case, the mechanism will solve the following equations to estimate the trustworthiness of f:

$$\omega(Y_f = y_i | Y_i = y_i) = \sum_{g \in \{1,2,...,K\}} T_f[g, y_f] \cdot \Big(\frac{T_i[g, y_i] \cdot P(g)}{\omega(Y_i = y_i)}\Big)$$

$\forall\, y_i, y_j \in \{1, 2, ..., K\}$

The coefficients $\frac{T_i[g,y_i] \cdot P(g)}{\omega(Y_i = y_i)}$ of this linear system don't depend on the answers given by worker f and the mechanism can determine in advance whether the system will be solvable by just looking at these coefficients. If $\omega(Y_i = y_i) \neq 0$ and the coefficient matrix is full rank, the answers of worker i are added. Note that we don't require answers of only truthful or high proficiency workers to be added to the pool. Answers from strategic workers can also be added to the answer pool as long as the pool expansion criterion is satisfied by the answers given by the workers.

6 Theoretical Properties

We will assume $N^{ij} \to \infty$ in all further discussion about the theoretical properties of the mechanism. We will later discuss the empirical performance of our mechanism while relaxing this assumption. It may be noted that this is **not** an assumption requiring every task to be solved by large number of workers but only that a worker and her respective peer solve many shared tasks.

Theorem 1. *The Deep Bayesian Trust Mechanism is dominant strategy incentive compatible for every worker and ensures strictly positive expected reward for the truthful strategy.*

Proof. As $N^{ij} \to \infty$, using lemma 1, the reward R_i of any worker i in the Deep Bayesian Trust Mechanism is given by :

$$\begin{aligned} R_i &= \Big(\sum_{k \in \{1,2,..,,K\}} T_i[k, k]\Big) - 1 \\ &= \Big(\sum_{k \in \{1,2,..,,K\}} \sum_{m \in \{1,2,..,,K\}} A_i[k, m] S_i[m, k]\Big) - 1 \end{aligned}$$

(Using Proposition 1)

For binary answer space ($K = 2$), this can be expanded as :

$$R_i = A_i[1, 1]S_i[1, 1] + A_i[1, 2]S_i[2, 1] + A_i[2, 1]S_i[1, 2] + A_i[2, 2]S_i[2, 2] - 1$$

Rearranging the terms,

$$R_i = \Big[A_i[1, 1]S_i[1, 1] + A_i[2, 1]S_i[1, 2]\Big] + \Big[A_i[1, 2]S_i[2, 1] + A_i[2, 2]S_i[2, 2]\Big] - 1$$

Assuming $A_i[1, 1] + A_i[2, 2] > 1$ (explained in remark 2 below), we get that $A_i[1, 1] > A_i[2, 1]$ and $A_i[2, 2] > A_i[1, 2]$.

Now, note that $\big[A_i[1,1]S_i[1,1]+A_i[2,1]S_i[1,2]\big]$ is a convex combination of $A_i[1,1]$ and $A_i[1,2]$ with $S_i[1,1]$ and $S_i[2,1]$ being the convex coefficients. Since $A_i[1,1] > A_i[2,1]$, this convex sum is maximized by using $S_i[1,1] = 1$ and $S_i[1,2] = 0$. A similar argument follows for the second independent term in the reward. Thus, the total reward is maximized by the identity strategy matrix (the truthful strategy). The reward in truthful strategy is thus,

$$R_i = A_i[1,1] + A_i[2,2] - 1$$

which is strictly positive.

For non-binary answer space ($K > 2$), the proof follows similarly assuming $A_i[g,g] > A_i[a,g],\ \forall a \neq g$. The reward for truthful strategy is $R_i = \sum_{k\in\{1,2,..,,K\}} A_i[k,k] - 1$ which is also strictly positive under the same assumption. □

Theorem 2. *In the Deep Bayesian Trust mechanism, a heuristic strategy results in zero expected reward.*

Proof. As $N^{ij} \to \infty$, using lemma 1, the reward R_i of any worker i in the Deep Bayesian Trust Mechanism is given by :

$$\begin{aligned} R_i &= \Big(\sum_{k\in\{1,2,..,,K\}} T_i[k,k]\Big) - 1 \\ &= \Big(\sum_{k\in\{1,2,..,,K\}} \sum_{m\in\{1,2,..,,K\}} A_i[k,m]S_i[m,k]\Big) - 1 \\ &= 1 - 1 = 0 \end{aligned}$$

We substituted $\Big(\sum_{k\in\{1,2,..,,K\}} \sum_{m\in\{1,2,..,,K\}} A_i[k,m]S_i[m,k]\Big)$ with 1 because S_i for a heuristic strategy has identical rows and S_i, A_i are row stochastic by definition. □

Remarks

1. The rewards only need to be scaled up by a constant to maintain the incentive properties when there is non-zero cost associated with the effort of solving the tasks. Similarly, when proficiency is a continuous increasing function of effort, increasing effort to get maximum proficiency remains the dominant strategy because reward (as shown in the proof of Theorem 1) is an increasing function of the proficiency.

2. The proof of Theorem 1 uses a natural assumption about the ability of workers ($A_i[1,1] + A_i[2,2] > 1$). Note that this is **not** an assumption on the honesty of the workers and is a fairly weak assumption even on their ability. The assumption merely ensures that the worker can obtain answers that are positively correlated with the ground truth. Such assumptions are common in all mechanisms [3]. Unlike other mechanisms, the assumption here only affects the best strategy of a given worker, not of all the workers. For example, if the assumption is not satisfied for a worker, she may find it better to deviate to a non-truthful strategy[1] but there is no effect on the best strategy for other workers.

3. If every single worker *irrationally* plays a heuristic strategy, then indeed our mechanism will not be able to expand its pool and force it to behave like other mechanisms which assign gold tasks to every worker. But the theoretical properties of the mechanism remain unaffected for *rational* workers even in such a degenerate case.

[1] Such informed deviation by a low proficiency worker to increase the accuracy of her reported answers is not really bad for the requester.

7 Fairness of Rewards

Recently, significant concerns have been raised about fairness and other ethical considerations while implementing algorithms that affect humans [1],[23]. The discussion on fair rewards in crowdsourcing has mostly been limited to issues such as minimum wages and adequate compensation for time and effort [28]. But there has not been any principled approach to address the issue of fairness in rewards from a non-discrimination perspective. For example, a worker with higher ability getting a lower reward than a worker with lower ability because of the difference in the way they were evaluated. The unfairness may be unintentional and random but may impact the perception of individual workers. Peer consistency mechanisms randomly select peers and reward the workers based on their answers and the answers of their respective peers. The reward of the workers is generally a function of their own ability as well as their peers' ability, thus making the rewards unfair. This unfairness issue in peer consistency mechanisms was also pointed out by [13]. The issue becomes more serious in mechanisms in which workers know in advance that they are being evaluated using peers with different proficiencies, for example in [5]. Our mechanism doesn't need to inform the workers about their peer but as we show, the mechanism can satisfy an even stronger definition of fairness.

Definition 9 (Fair Reward Mechanism)**.** *A reward mechanism is called fair if the expected reward of any worker is directly proportional to the accuracy of the answers reported by her and independent of other environmental variables.*

Here, the environmental variables include the factors such as abilities or honesty levels of co-workers and don't include any scaling constants that may be applied identically on the rewards of all workers. The scaling constants are necessary for satisfying budget constraints or cost compensation.

This is a reasonable definition of fairness and is in agreement with the broader theory for ***individual fairness*** of algorithms. For example, the pioneering work of [7] defines that fair algorithms take similar decisions for individuals with similar relevant attributes. The relevant attribute in our case is the worker's accuracy. The definition is also non-trivial to satisfy. In existing peer consistency based mechanisms, the rewards also depend on the unknown ability of the peer (even if the peer can be believed to be truthful). For example, in the mechanism of [3], the reward is an increasing function of the proficiencies of both the worker and her peer.

Theorem 3. *The Deep Bayesian Trust Mechanism is fair.*

Proof. The proof follows from Lemma 1, which ensures that reward of worker i converges in the limit to

$$\Big(\sum_{k\in\{1,2,..,,K\}} T_i[k,k]\Big) - 1$$

By definition, $T_i[k,k], \forall k \in \{1,2,...,K\}$ measure the accuracy of the answers reported by the worker. Thus, even though the mechanism has only estimates of the accuracy of the workers' answers and the estimates are indeed obtained using the answers of the peers and their trustworthiness but the consistency property of these estimates ensures the asymptotic fairness of the mechanism. □

This is perhaps a surprising result because in the existing framework of the peer consistency mechanisms, one would reason that it is impossible for the rewards to not depend on the accuracy of the peer unless the peer is always correct.

8 Experiments

In this section, we examine the performance of our mechanism empirically. We simulate the settings in which workers with different proficiencies A_i report answers to different tasks. The proficiency matrices of different workers were generated independently such that the diagonal entries $A_i[k,k]\ \forall k \in \{1,2,...,K\}$ were $\beta(5,1)$ distributed. In another setting, $A_i[k,k]\ \forall k \in \{1,2,...,K\}$ were uniformly distributed in $(\frac{1}{K},1]$. Note that the entries $A_i[k,k]\ \forall k \in$

$\{1, 2, ..., K\}$ for a given worker i are not necessarily the same as they are also independently generated. Rest of the entries are generated randomly such that every row of proficiency matrix sums to 1.

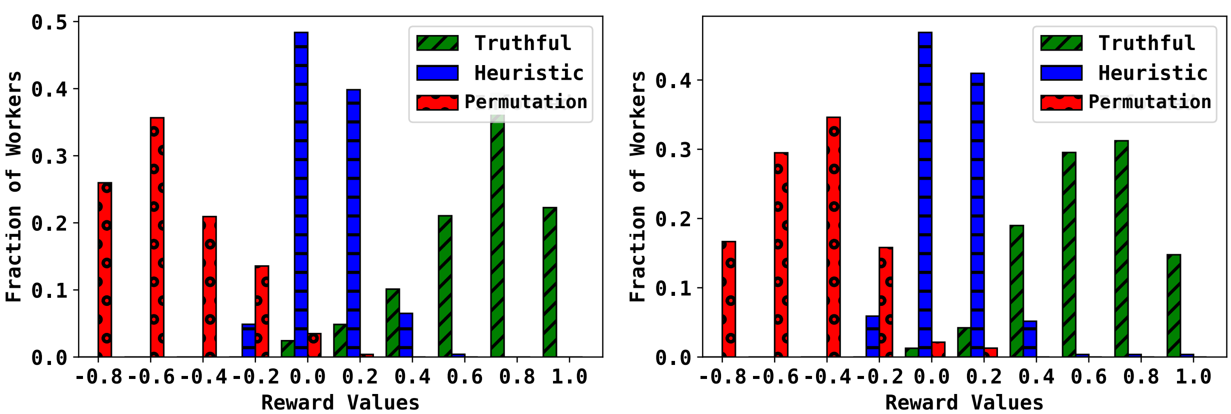
We consider following strategies that workers may play :

1. **Truthful**- Workers obtain answer for any given task based on their respective proficiency matrices and report the answers truthfully.

2. **Heuristic** - Workers' reported answers are generated independently of their proficiency using a common distribution over the answer space.

3. **Permutation** - Workers obtain answer for any given task based on their respective proficiency matrices but they apply a common permutation on the answers before reporting it to the mechanism. For example, in a ternary answer space, a permutation f can be as follows : $f(1) = 2, f(2) = 3, f(3) = 1$, i.e., whenever the obtained answer is 1, workers report 2, for 2, they report 3 and for 3, they report 1. In a binary answer space, this corresponds to reporting the opposite of the obtained answer.

In general, the simulations performed in the literature for peer based mechanism compare the average reward in different equilibria. For example, the reward statistics of workers when everyone uses a truthful strategy may be compared with the reward statistics when everyone uses a heuristic strategy. But our stronger theoretical result (dominant strategy incentive compatibility) demands stronger simulations also. We go beyond comparing just equilibrium rewards and instead compare the rewards of workers playing different strategies against one another at the same time. More precisely, in our simulations, we don't require every worker to play same strategy. Any worker can play a heuristic, permutation or truthful strategy with equal probability. Such simulations can't be handled by mechanisms that guarantee only Nash equilibrium results[2]. We show that in our mechanism, there is a clear distinction between the rewards of workers playing different strategies with truthful workers being nicely rewarded and others being penalized.



(a) Proficiency Distribution : Beta

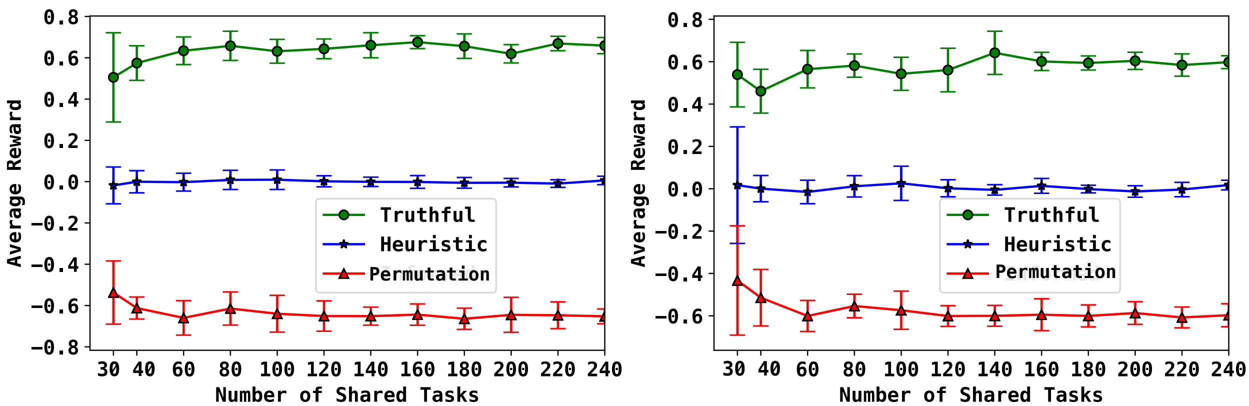
(b) Proficiency Distribution : Uniform

Figure 1: Distribution of rewards for workers with different proficiencies playing different strategies

Workers were simulated to be hired in 4 rounds, with 5, 25, 125 and 625 workers in successive rounds. Figure 1a compares the distribution of rewards of workers playing the three strategies in the first setting ($\beta(5, 1)$ distributed

[2]Truthful equilibrium makes truthtelling the best response if others don't deviate.

proficiencies). Clearly, the rewards of workers playing the heuristic strategy are centered around 0, as expected from Theorem 2. The reward of workers playing truthful strategy are centered around a strictly positive value as predicted by Theorem 1. On the contrary, the rewards of workers playing the permutation strategy are symmetrically centered around a strictly negative value. It may be noted that in existing peer consistency mechanisms, permutation strategies (in equilibria) are equally profitable as the truthful strategy, which is not the case with our mechanism. Figure 1b compares the reward distribution in the second setting (uniformly distributed proficiencies). Observations similar to the prior setting are made in this setting as well. The difference is that for the truthful strategy, rewards in the former settings are slightly more skewed towards the positive side. It is an expected observation because the rewards of truthful workers are shown to be an increasing function of their proficiencies in the proof of Theorem 1 and the proficiency distribution in $\beta(5, 1)$ is indeed skewed.



(a) Proficiency Distribution : Beta

(b) Proficiency Distribution : Uniform

Figure 2: Average reward of workers playing different strategies under different number of shared tasks

We now show the **robustness of our technique w.r.t. the number of shared tasks between workers**. We discussed the asymptotic properties of the mechanism earlier. Hence, the simulation study is important to show the performance the mechanism for a finite number of shared tasks in practice. Figure 2a compares the average of rewards of the workers (with $\beta(5, 1)$ distributed proficiencies) playing different strategies under different settings of the number of shared tasks. The trend shown in previous simulation can be observed to be very robust to the number of shared tasks. Error bars show the standard deviation in 10 repeated runs. As we observe empirically, the reward mechanism is attractive even when the number of shared tasks is not large. This also means that with only 30 gold tasks (and given only to 5 workers), the mechanism can reward $5 + 25 + 125 + 625 + ...$ workers. Figure 2b compares the average of rewards of the workers (with uniformly distributed proficiencies) playing different strategies under different settings of the number of shared tasks. The results are similar to that for $\beta(5, 1)$, further attesting the claim.

9 Amazon Mechanical Turk Study

Any conclusive study on the effects of reward mechanisms requires a large budget and needs to be conducted over a long period. Such a study is beyond the scope of this paper. We performed a limited scale study on Amazon Mechanical Turk to make some preliminary observations about the ease of implementing the mechanism and workers' response to the mechanism. We created some synthetic tasks which resemble tasks requiring human intelligence (natural language understanding) and elicited the answers of the crowd on MTurk with and without our reward mechanism in place. The advantage of using synthetically generated tasks was that we had access to ground truth and could judge the performance of workers with and without our mechanism objectively. The structure of our task (named *'story disentanglement task'*) was the following : we mixed a few sentences from different real news stories into one paragraph and asked the workers to count the number of news stories in the paragraph. Solving this task requires identifying the context of different sentences and whether they are related. The number of sentences in paragraphs was kept independent of the number of stories, making it harder to guess by just looking at the paragraph length. We asked workers to give a binary answer ('Yes' if the number of stories is less than 3 and 'No' otherwise). We also asked them to identify the sentences belonging to different stories. We will discuss only the binary answers of the workers. Each HIT corresponds to giving an answer for one such paragraph. We conducted the experiments under two settings.

- In the first setting, workers were told that they would be paid 0.03\$ per HIT and there would be additional performance based payments without discussing a specific reward rule. We will refer to this setting as the **unspecified** reward setting.
- In the second setting, each HIT was worth 0.03\$ and we explained our Deep Bayesian Trust mechanism to the workers in plain English with almost no use of mathematical language or notations. We will refer to this setting as the **DB Trust** setting.

In the DB Trust setting, batches of 80 HITs were designed such that each batch had 40 HITs in common with another batch to satisfy peer relationship. In both settings, we had 3 workers giving answers for each paragraph, giving us a total 3×480 HITs from 480 paragraphs. We thus collected a dataset of 1440 worker responses on these HITs, 720 in each setting. In total, 129 workers participated in the experiment. We judge the mechanism on two most important criteria. First, the ability to discourage workers from heuristic reporting and second, the ability to get more accurate answers from crowd.

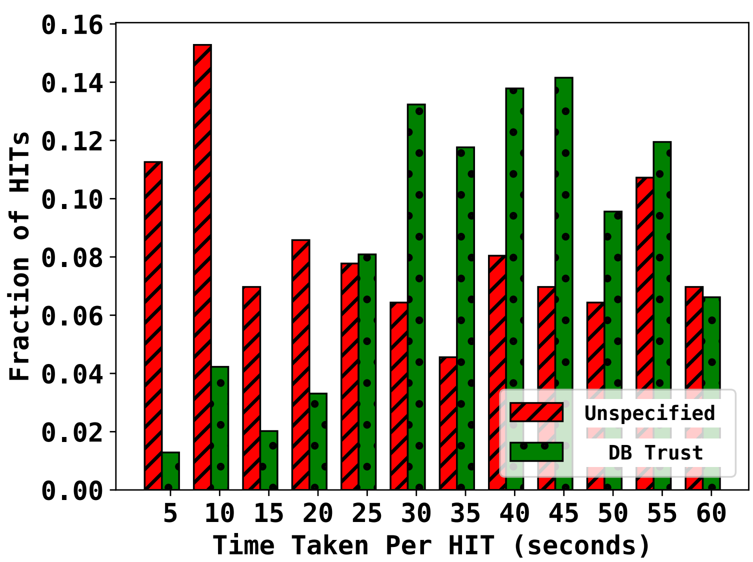


Figure 3: Time Spent on HITs

Observations

1. Figure 3 compares the time workers spent on solving the tasks in the two settings. The fraction of HITs that were given very little time has significantly decreased with our mechanism and the fraction of HITS that were given

more time has significantly increased (the green distribution with dots is more skewed towards the right side as compared to the red distribution with slashes, which is more skewed towards the left). This can be interpreted [10] as a success in eliciting effort from the crowd and discouraging low quality/heuristic reporting. We used a browser based JavaScript solution to measure the actual time spent on solving tasks to get tight estimates of time spent in the DB Trust setting, without workers being aware of it. Amazon uses the difference between time of accepting and submitting a HIT as estimates of time spent, which (even after filtering very large values) tend to be highly inflated. As one can see, even with such tight estimates in the DB Trust setting, the time spent by workers is better.

2. The average accuracy of workers was found to increase from 70.86% in the unspecified setting to 79.17% in the DB Trust setting.

3. The average accuracy of all responses was also found to increase from 75.69% to 79.17% with our scheme.

10 Conclusions

In this paper, we discussed the problem of quality control in crowdsourcing. To address the root of the problem, it is necessary to design reward schemes that align the incentives of the workers and encourage them to provide high quality data. We proposed the Deep Bayesian Trust mechanism to achieve this in large scale crowdsourcing. The mechanism rewards the workers for the correctness of their reports without checking every worker with gold standard tasks. Instead, it uses the correlation in the answers of the workers and peers to estimate their accuracy. The mechanism is game theoretically sound and robust to any strategic manipulation. Workers strictly maximize their reward by putting their best effort and reporting truthfully. The mechanism also ensures fair reward to workers. The reward of any worker is independent of the accuracy of other workers and depends solely on her own accuracy, thus contributing towards the bigger movement of making algorithmic decisions fair and ethical. The Deep Bayesian Trust mechanism is easy to implement in practice and can interact with crowdsourcing platforms through their APIs without manual intervention of the requester. We demonstrated the properties of the mechanism through simulations in settings of practical interest. We show that the mechanism effectively rewards the truthful workers and penalizes the non-truthful ones, even when workers with different abilities and strategies act at the same time. We further studied the response of real crowdworkers (from Amazon Mechanical Turk) towards our scheme and found that it can encourage workers to spend more time on the tasks and leads to better average accuracy of their responses.

References

[1] ACM. Statement on algorithmic transparency and accountability. 2017.

[2] Arpit Agarwal, Debmalya Mandal, David C. Parkes, and Nisarg Shah. Peer prediction with heterogeneous users. In *Proceedings of the 18th ACM Conference on Economics and Computation (EC-2017)*, 2017.

[3] Anirban Dasgupta and Arpita Ghosh. Crowdsourced judgement elicitation with endogenous proficiency. In *Proceedings of the 22nd international conference on World Wide Web*, pages 319–330. ACM, 2013.

[4] Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics*, pages 20–28, 1979.

[5] Luca de Alfaro, Marco Faella, Vassilis Polychronopoulos, and Michael Shavlovsky. Incentives for truthful evaluations. *arXiv preprint arXiv:1608.07886*, 2016.

[6] Djellel Eddine Difallah, Gianluca Demartini, and Philippe Cudré-Mauroux. Mechanical cheat: Spamming schemes and adversarial techniques on crowdsourcing platforms. In *WWW, CrowdSearch Workshop*, pages 26–30, 2012.

[7] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 214–226. ACM, 2012.

[8] Alice Gao, James R Wright, and Kevin Leyton-Brown. Incentivizing evaluation via limited access to ground truth: Peer-prediction makes things worse. In *2nd Workshop on Algorithmic Game Theory and Data Science at EC 2016.*, 2016.

[9] Xi Alice Gao, Andrew Mao, Yiling Chen, and Ryan Prescott Adams. Trick or treat: putting peer prediction to the test. In *Proceedings of the fifteenth ACM conference on Economics and Computation*, pages 507–524. ACM, 2014.

[10] Chien-Ju Ho, Aleksandrs Slivkins, Siddharth Suri, and Jennifer Wortman Vaughan. Incentivizing high quality crowdwork. In *Proceedings of the 24th International Conference on World Wide Web*, pages 419–429. ACM, 2015.

[11] Zehong Hu, Yang Liu, Yitao Liang, and Jie Zhang. A reinforcement learning framework for eliciting high quality information. 2018.

[12] Panagiotis G Ipeirotis, Foster Provost, and Jing Wang. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pages 64–67. ACM, 2010.

[13] Ece Kamar and Eric Horvitz. Incentives for truthful reporting in crowdsourcing. In *Proceedings of the 11th international conference on autonomous agents and multiagent systems-volume 3*, pages 1329–1330. International Foundation for Autonomous Agents and Multiagent Systems, 2012.

[14] David R Karger, Sewoong Oh, and Devavrat Shah. Iterative learning for reliable crowdsourcing systems. In *Advances in Neural Information Processing Systems*, pages 1953–1961, 2011.

[15] Aniket Kittur, Ed H Chi, and Bongwon Suh. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 453–456. ACM, 2008.

[16] Edith Law, Ming Yin, Joslin Goh, Kevin Chen, Michael A Terry, and Krzysztof Z Gajos. Curiosity killed the cat, but makes crowdwork better. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 4098–4110. ACM, 2016.

[17] Yaliang Li, Jing Gao, Chuishi Meng, Qi Li, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. A survey on truth discovery. *SIGKDD Explor. Newsl.*, 17(2):1–16, 2016.

[18] Siyuan Liu, Chunyan Miao, Yuan Liu, Han Yu, Jie Zhang, and Cyril Leung. An incentive mechanism to elicit truthful opinions for crowdsourced multiple choice consensus tasks. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2015 IEEE/WIC/ACM International Conference on*, volume 2, pages 96–103. IEEE, 2015.

[19] Panagiotis Mavridis, David Gross-Amblard, and Zoltán Miklós. Using hierarchical skills for optimized task assignment in knowledge-intensive crowdsourcing. In *Proceedings of the 25th International Conference on World Wide Web*, pages 843–853, 2016.

[20] Nolan Miller, Paul Resnick, and Richard Zeckhauser. Eliciting informative feedback: The peer-prediction method. *Management Science*, 51(9):1359–1373, 2005.

[21] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Advances in Neural Information Processing Systems*, pages 1196–1204, 2013.

[22] David Oleson, Alexander Sorokin, Greg P Laughlin, Vaughn Hester, John Le, and Lukas Biewald. Programmatic gold: Targeted and scalable quality assurance in crowdsourcing. *Human computation*, 11(11), 2011.

[23] John Podesta, Penny Pritzker, Ernest J. Moniz, John Holdren, and Jefrey Zients. Seizing opportunities and preserving values. *Executive Office of the President*, 2014.

[24] Dražen Prelec. A bayesian truth serum for subjective data. *science*, 306(5695):462–466, 2004.

[25] Goran Radanovic and Boi Faltings. A robust bayesian truth serum for non-binary signals. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence (AAAI" 13)*, number EPFL-CONF-197486, pages 833–839, 2013.

[26] Goran Radanovic, Boi Faltings, and Radu Jurca. Incentives for effort in crowdsourcing using the peer truth serum. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 7(4):48, 2016.

[27] Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *Journal of Machine Learning Research*, 11(Apr):1297–1322, 2010.

[28] Florian Alexander Schmidt. The good, the bad and the ugly: Why crowdsourcing needs ethics. In *Cloud and Green Computing (CGC), 2013 Third International Conference on*, pages 531–535. IEEE, 2013.

[29] Victor Shnayder, Arpit Agarwal, Rafael Frongillo, and David C. Parkes. Informed truthfulness in multi-task peer prediction. EC '16, pages 179–196. ACM, 2016.

[30] Bo Waggoner and Yiling Chen. Output agreement mechanisms and common knowledge. In *Second AAAI Conference on Human Computation and Crowdsourcing*, 2014.

[31] Jens Witkowski and David C Parkes. A robust bayesian truth serum for small populations. In *AAAI*, 2012.

[32] Ming Yin, Yiling Chen, and Yu-An Sun. The effects of performance-contingent financial incentives in online labor markets. In *AAAI*, 2013.

[33] Yudian Zheng, Jiannan Wang, Guoliang Li, Reynold Cheng, and Jianhua Feng. Qasca: a quality-aware task assignment system for crowdsourcing applications. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1031–1046. ACM, 2015.