

Incentives for Truthful Evaluations

Luca de Alfaro Michael Shavlovsky Vassilis Polychronopoulos
luca@ucsc.edu, {mshavlov, vassilis}@soe.ucsc.edu
Computer Science Department
University of California
Santa Cruz, CA, 95064, USA

Marco Faella
m.faella@unina.it
Dept. of Electrical Engineering and Information Technologies
University of Naples Federico II, Italy

Technical report UCSC-SOE-16-14
August 28, 2016

Abstract

We consider crowdsourcing problems where the users are asked to provide evaluations for items; the user evaluations are then used directly, or aggregated into a consensus value. Lacking an incentive scheme, users have no motive in making effort in completing the evaluations, providing inaccurate answers instead. We propose incentive schemes that are truthful and cheap: truthful as the optimal user behavior consists in providing accurate evaluations, and cheap because the truthfulness is achieved with little overhead cost. We consider both discrete evaluation tasks, where an evaluation can be done either correctly, or incorrectly, with no degrees of approximation in between, and quantitative evaluation tasks, where evaluations are real numbers, and the error is measured as distance from the correct value. For both types of tasks, we propose hierarchical incentive schemes that can be effected with a small amount of additional evaluations, and that scale to arbitrarily large crowd sizes: they have the property that the strength of the incentive does not weaken with increasing hierarchy depth. Interestingly, we show that for these schemes to work, the only requisite is that workers know their place in the hierarchy in advance.

1 Introduction

Crowdsourcing allows access to large populations of human workers, and it can be an efficient and cheap solution for many applications. The very farming out of work to many independent workers, however, creates the problem of quality control. In the absence of effective supervision or quality-control mechanisms, the workers may submit low quality work, or they may deliberately engage in straight-out vandalism. Workers can also collude with each other to game the system and collect rewards without performing the required work. In this paper, we describe supervisory schemes that provide an incentive towards high-quality work, and we show that the incentive is both cheap in terms of the required supervisor time and work overhead, and effective in making honest and accurate work the best strategy for workers.

We focus on crowdsourcing tasks which are *verifiable*, that is, they have objective answers that a supervisor can check to conclude whether a worker is submitting quality work or not. We further consider two types of verifiability: *binary*, and *quantitative*. In binary verifiable tasks, the question of whether a worker submits quality work can be answered with either a Yes or a No. In quantitatively verifiable tasks, the quality question can be answered only quantitatively as a measure of distance between the work submitted by

a worker, and the work that was expected. Classification tasks are examples of binary verifiable tasks, as supervisors can check that the classification in discrete categories submitted by a worker matches expectations. Grading tasks are examples where quantitative evaluation is natural; the quality of the human worker can be determined as a function of the distance of the worker’s answer to the true grade.

We propose schemes that provide truthful incentives to the human workers at a low cost for the supervisors regardless of the size of the human worker population. We first study a simple *one-level* scheme and thereafter we propose a *hierarchical* scheme.

In the simple *one-level* scheme, workers perform tasks which are directly evaluated by the supervisor with some probability. We study the conditions that ensure that workers maintain the incentive to provide truthful answers. However, the one-level scheme does not scale to large crowds, as the supervisor needs to perform a prohibitively high amount of work to impose truthful incentives.

The novel *hierarchical* scheme we propose can scale to arbitrarily large crowds of human workers. The mechanism organizes workers in a hierarchy, where the supervisor is at the top, and the other workers are arranged in *layers* below. Our hierarchical schemes provide a truthful incentive to the workers regardless of their level in the hierarchy. Furthermore, the tasks assigned to workers high in the hierarchy are of the same kind as the tasks performed by lower ranked workers. In particular, we do not need to split workers into “regular” workers and meta-reviewers. As the worker population increases, the hierarchy becomes deeper, but the amount of work that the supervisor needs to do remains constant, and so does the incentive towards correct behavior. We show that the only information about the hierarchy that needs to be communicated beforehand to the workers is their level in the hierarchy itself. We provide matching upper and lower bounds for the amount of information that needs to be communicated beforehand to workers in the hierarchy to maintain a truthful incentive, showing that a logarithmic amount of information in the number of workers is both necessary and sufficient.

We study the practical aspects of the implementation of the hierarchy. Many crowdsourcing tasks benefit from redundancy, that is, from assigning the same task to more than one worker. For instance, by assigning the same item to multiple graders, it is possible to reconstruct a higher-accuracy grade for the item than would be available from one grader alone [33]. We show that in redundant tasks in which there is no control over task allocation to workers, the problem of creating an optimal hierarchy is NP-hard. We present fast approximation algorithms that are optimal within constant factors. If the supervisor can control the allocation of tasks to workers, as in many real applications, we show that constructing the hierarchy is an easy problem.

We developed our results first in the simpler case with binary verifiable tasks. These are common in classification tasks: spam or not, correct answer or not, etc. We analyzed a model in which workers either behave correctly, or make mistakes, and have a fixed effort cost to complete a task. Then we showed how the results directly transfer to the case of quantitative tasks. The difference being that the notion of a *correct* task is replaced by the notion of a *correct within ε* task.

Our analysis is based on a general model that describes workers’ performance using a general function that gives the user effort required to achieve a desired level of variance. The model is general since the only assumption about the function is that it is monotonically decreasing; the higher the variance the less effort is required, or reversely, workers have to make the highest effort to achieve the minimum variance. This describes a realistic model where increased worker effort produces higher expected precision of the worker’s answers, and is similar to other models proposed in the literature [28].

We show that in our hierarchical schemes, the precision factor ε is constant across the hierarchy, and does not degrade with increasing depth of the workers. In other words, hierarchical distance from the supervisor does not entail less precision in the tasks performed.

The proposed schemes are thus applicable to a multitude of crowdsourcing applications, from conventional classification tasks using generic crowds in crowdsourcing marketplaces to peer grading in Massive Open Online Courses with an arbitrarily large population of students.

2 Related Work

Providing incentives to human agents to return truthful responses is one of the central challenges for crowdsourcing algorithms and applications [8].

Prediction markets are models with a goal of obtaining predictions about events of interest from experts. After experts provide predictions, a system assigns a reward based on a scoring rule to every expert. Proper scoring rules ensure that the highest reward is achieved by reporting the true probability distribution [21, 9, 4]. An assumption of the scoring rules is that the future outcome must be observable. This assumption prevents crowdsourcing systems to scale to large crowds as obtaining the correct answer for each event or task is prohibitively expensive.

The model presented in [3] relaxes this assumption. The proposed scoring rule evaluates experts by comparing them to each other. The model assigns a higher score for an expert if her predictions are in agreement with predictions of other experts. Work [3] belongs to the class of peer prediction methods. Peer prediction methods is wide class of models for providing incentives [18, 16, 26, 27, 25, 5, 12, 29, 31, 32, 24, 34, 35]. Such methods elicit truthful answers by analyzing the consensus between workers in one form or another. Peer prediction methods encourage cooperation between workers and, as a result, promote uninformative equilibria. The study in [10] shows that for the scoring rules proposed in the peer-prediction method [16], a strategy that always outputs “good” or “bad” answer is a Nash equilibrium with a higher payoff than the truthful strategy. Works by [12, 24] show that the existence of such equilibria is inevitable. In contrast, hierarchical incentive schemes we propose make the truthful strategy the only Nash equilibrium.

The model described in [12] considers a scenario of rational buyers who report on the quality of products of different types. In the developed payment mechanism the strategy of honest reporting is the only Nash equilibrium. However, the model requires that the prior distribution over product types and condition distributions of qualities is the common knowledge. This requirement is a strong assumption.

The Bayesian Truth Serum scoring method proposed in [18] elicits truthful subjective answers on multiple choice questions. The author shows that the truthful reporting is a Nash equilibrium with the highest payoff. The model is different from other approaches in that besides the answers, workers need to provide predictions on the final distribution of answers. Workers receive a high score if their answer is “surprisingly” common - the actual percentage of their answer is larger than the predicted fraction. Similarly, incentive mechanisms in [27, 25, 29, 30, 31] require workers provide belief reports along with answers on tasks. Truthful mechanisms in [16, 32, 35] requires knowledge about the distribution from which answers are drawn. Our mechanisms do not rely on worker’s beliefs on other workers’ responses nor require knowledge about the global answer distribution.

The work in [2] studies the problem of incentives for truthfulness in a setting where persons vote other persons for a position. The analysis derives a randomized approximation technique to obtain the higher voted persons. The technique is strategyproof, that is, voters (which are also candidates) cannot game the system for their own benefit. The setting of this analysis is significantly different from ours, as the limiting assumption is that the sets of voters and votees are identical. Also, the study focuses on obtaining the top- k voted items, while in our setting we do not necessarily rank items.

The PeerRank method proposed in [20] obtains the final grades of students using a fixed point equation similar to the PageRank method. However, while it encourages precision, it does not provide a strategyproof method for the scenario that students collude to game the system without making the effort to grade truthfully.

Using golden sets is a practice for quality control in crowdsourcing [36]. Golden sets are designated sets of tasks used to evaluate performance of workers; such sets showed a positive impact on worker performance in crowdsourcing systems [37]. Golden sets, however, can be difficult and costly to obtain [38], as they create an overhead for workers to perform extra tasks for quality control. Moreover, golden sets might be not available in advance. For example, in peergrading of homework assignments the total set of homework

submissions cannot be obtained before the homework is posted. Also, information on the competence of workers from previous homeworks or classrooms cannot be used reliably for newer homeworks or in different classrooms with potentially different material. The incentive schemes we propose do not require golden sets.

3 The Binary-verifiable Model

We make the binary-verifiable setting precise via the following model. Let U and I be the set of workers and tasks respectively. Every worker u performs a subset of tasks from I . Workers have a choice on how to perform a task. They can either complete the task or they can defect and not do the task. We construct a bipartite graph $G = (U \cup I, E)$ with tasks and workers as nodes. For a task $i \in I$ and a worker $u \in U$, the edge (i, u) belongs to the set of edges E iff the worker u was assigned the task $i \in I$. We denote the set of tasks assigned to a worker u as ∂u , the set of workers assigned with a task i as ∂i .

Strategies. A worker can have two pure strategies: one is to be truthful and the other one is to defect. Apart from the pure strategies, there is a continuum of mixed strategies that are random choices between being truthful or defecting. Workers with the truthful strategy diligently perform tasks. We assume that correctly performing a task has cost C , which for simplicity we assume as constant, independent from the specific task or worker. Defecting on a task has cost 0. Defectors, however, can be assigned a penalty if reviewed by the supervisor.

Supervision. We assume that there is a supervisor that can verify whether tasks are done correctly or not. When the supervisor reviews the completed task of a worker, if the worker has defected on that task then the supervisor assigns a penalty P , otherwise the supervisor assigns no penalty to the worker. As for workers, we assume that the cost for the supervisor of verifying a task is the same for all tasks. We also assume that by verifying a task i , the supervisor is able to verify all workers in ∂i . This means, essentially, that all workers assigned i would have yielded the *same* answer when doing i correctly. This is a reasonable model for classification tasks, as well as for individual tasks which do not benefit from being done by multiple workers simultaneously.

If the supervisor verifies all the workers, then every worker has an incentive to perform a task, as long as the penalty P is greater than worker cost mC , where m is the number of tasks that are asked of the worker. Our goal is to develop supervision schemes that minimize the supervisor’s work while providing an incentive for workers to be truthful.

We will propose one level and multiple levels or hierarchical supervision schemes that provide incentive to workers so that they play with the truthful strategy. In the one level scheme a worker’s penalty depends only on the supervisor. In the hierarchical scheme a worker’s penalty depends either on the supervisor or on other workers.

4 One Level Supervised Schemes

In this section we study “one-level”, or flat, supervised schemes where all workers have the same role.

To verify a worker $u \in U$, the supervisor examines a task $i \in \partial u$ assigned to the worker. If the work defected on the task i , then the supervisor assigns penalty P to worker u . To provide an incentive to workers U to play with the truthful strategy. the supervisor chooses a subset of tasks to examine. We use p to denote a probability that a randomly chosen worker has a task that belongs to the subset. The higher the probability p , the higher the influence of the supervisor on all workers.

According to our model assumptions, by examining a tasks $i \in I$, the supervisor evaluates all workers ∂i assigned to the task i . We will separately consider two crowdsourcing settings: one without redundancy where every task $i \in I$ has only one worker and one with redundancy where a task $i \in I$ has more than one worker assigned to it.

4.1 Crowdsourcing without Redundancy

We first study the case when every task $i \in I$ has exactly one worker assigned to it. To verify the work done, the supervisor randomly chooses a subset of items so that no two items with the same worker belong to the subset, and uses the work performed on these items to assign penalties to defecting workers. By selecting first a random subset of m workers, and then picking an item for each worker, the supervisor can ensure a probability $p = m/|U|$ of verifying a worker. Theorem 1 establishes a lower bound on p so that workers have an incentive to be truthful. To prove it, we show the following easy proposition first.

Proposition 1. *If $kC \neq pP$, then for every mixed strategy of a worker with k tasks there is a pure strategy that dominates it.*

Proof. Consider a mixed strategy when a worker is being truthful with probability $r \in (0, 1)$ and defects with probability $1 - r$. The expected loss of the worker is

$$l = rkC + (1 - r)pP ,$$

where k is the number of tasks performed by the worker. If $kC > pP$ then the workers can minimize the loss by setting $r = 0$, that is, defecting. Conversely, if $kC < pP$ then then loss is minimized at $r = 1$. \square

Theorem 1. *If every worker is assigned at most K tasks, and the probability p satisfies the inequality*

$$p > \frac{KC}{P} , \tag{1}$$

then the truthful strategy has the smallest loss.

Proof. Due to Proposition 1 we can limit our attention to the truthful and defecting strategies which both pure strategies. The loss of the truthful strategy is $l_T = KC$, the loss of the defecting strategy it $l_P = pP$. The truthful strategy has the smallest loss when $l_P > l_T$, or $p > \frac{KC}{P}$. \square

The number of workers $m = p|U|$ that the supervisor needs to examine grows with the total number of workers. This limits the applicability of the flat approach to relatively small task sets. In a subsequent section, we will develop a hierarchical setting that overcomes this limitation.

4.2 Crowdsourcing with Redundancy

In this section we study the case when a task has multiple workers assigned to it. This is useful when, by comparing the ways in which the tasks have been performed, it is possible to produce higher-accuracy estimates of the solution. When a task is performed by multiple workers, verifying a single task i can be used to verify all the users in ∂i . The supervisor can leverage this in order to try to minimize the number of tasks to be verified, while guaranteeing a worker verification probability p that satisfies Theorem 1. We will show that when a graph G of tasks and workers is given, i.e. when we do not have control over task allocation, then constructing the smallest subset S is \mathcal{NP} -hard. However, if we can control tasks allocation, then we can easily construct graphs on which the set of tasks that need verification is as small as possible, in a sense that will be made precise.

4.2.1 The assignment graph is given.

We first study a scenario in which the worker-task assignment is fixed, and we must choose the subset $S \subseteq I$ of tasks verified by the supervisor. When the supervisor examines a task $i \in I$, she evaluates all the workers ∂i who were assigned to the task i . Figure 1 illustrates a case when examining 3 tasks is enough to evaluate all the workers.

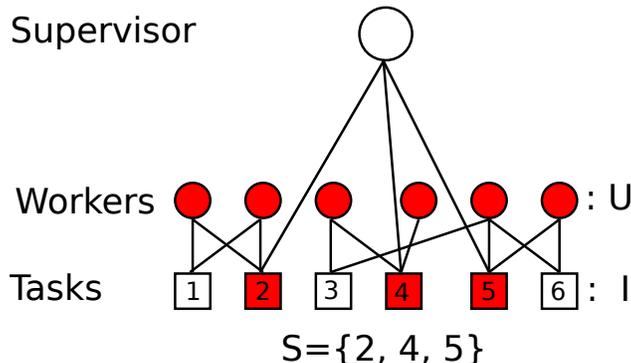


Figure 1: An example of a graph with all 6 workers being evaluated based on a set of 3 tasks. The supervisor inspects tasks 2, 4 and 5 that connected to all workers. The tasks and workers the supervisor reaches out are colored.

The supervisor wants to spend the least amount of effort to evaluate at least m workers. For the case $p = 1$, or $m = |U|$, the supervisor needs to find the smallest subset of tasks such that every worker is assigned one task from the subset. We name the problem of finding such a set the *Superior Assignment* problem (abbreviated to *SA*). The following theorem shows that the *Supervisor Assignment* Problem is \mathcal{NP} -hard.

Theorem 2. *Supervisor Assignment problem is \mathcal{NP} -hard.*

Proof. We will show that finding the smallest vertex cover for any graph is an instance of the Supervisor Assignment problem. Thus, solving the *Supervisor Assignment* problem is at least as hard as solving Vertex Cover.

Let $G = (V, E)$ be an arbitrary graph with vertexes V and edges E . We construct a bipartite revision graph G' for a set of users U and set of items I by taking $U = E$ and $I = V$: that is, we use users in our bipartite graph to represent the edges of the original graph. Each user $u \in U$ is assigned to review items v_1, v_2 , where in G the edge u connects v_1 and v_2 . The graph G' is called the incidence graph [22]. It is immediate to see that a subset of vertices $V' \subseteq V$ is a *vertex cover* for G if and only if picking all items in V' enables the verification of all users E of G' . Thus, Vertex Cover can be reduced to the Supervisor Assignment problem. \square

We now show that, if every worker is assigned at most k tasks, there are fast k -approximation algorithm for SA. A k -approximation algorithm finds a subset S' of tasks such that $|S'| < k|S|$, where S is the optimal solution.

We will show that the SA problem on graph G is equivalent to the VC problem on a hypergraph with edge size at most k . A hypergraph $H = (V, F)$ is a set of vertices V and hyperedges F . A hyperedge $f \in F$ connects a subset of edges from V . Hypergraph H has edge size at most k if every edge $f \in F$ contains at most k nodes. There are known simple k -approximation algorithms for VC on k -bounded hypergraphs [23].

Proposition 2. *The Supervisor Assignment problem for a bipartite review graph $G = (U \cup I, E)$ with degree at most k is equivalent to Vertex Cover for a hypergraph with edge size at most k .*

Proof. The SA problem is immediately equivalent to a VC cover for a hypergraph that has U as vertex set, and has I as edge set, where each edge $i \in I$ connects the vertices that correspond to the workers to which i is assigned. \square

A simple k -approximation algorithm works as follows. Let $G = (V \cup F, E)$ be a bipartite graph and $S = \emptyset$. While set E is not empty, we randomly choose an edge $(v, f) \in E$, add node v to set S , and delete all edges incident to v or f . When E is empty, set S is a k -approximation to the SA problem on graph G .

4.2.2 The assignment graph can be constructed.

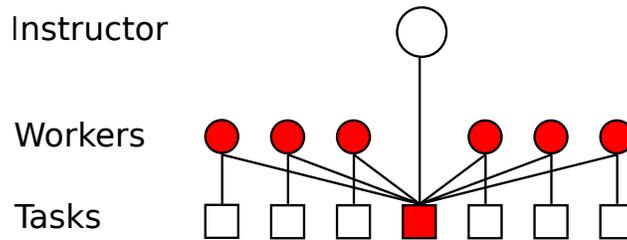


Figure 2: An example of review graph where the supervisor evaluates all workers by reviewing only one task.

We showed above that if the assignment graph is given, optimally choosing the items is a hard problem. However, if we can construct the assignment graph, then it is easy to ensure optimality. One extreme example is depicted on figure 2. All workers have only one task in common. The supervisor can evaluate all the workers by verifying only one task. From a crowdsourcing perspective, however, concentrating effort of all workers on one assignment has the unwelcome effect that all other tasks receive fewer workers. If we use worker multiplicity for a task in order to achieve higher reliability in the solution of a task, this is undesirable. A natural assumption is to require the review graph G to be k -regular.

To construct a k -regular review graph, we proceed as follows. We select $n = \lceil \frac{|U|}{k} \rceil$ “peg” tasks first. Each of these peg tasks will be done by a set of k non-overlapping workers, so by verifying the n peg tasks, the supervisor is able to verify all workers ($p = 1$). For smaller values of the verification probability p , the supervisor can simply choose to verify a randomly chosen subset of the peg tasks. We assume, of course, that the workers cannot compare their work with each other, so that they cannot infer which tasks are the peg tasks among those they are assigned. Once the peg tasks and their reviewers are chosen, we assign the other tasks to workers in any way that leads to k -regularity. It is easy to see that this construction is optimal, for $|U|$ workers doing k tasks each cannot be verified by picking fewer than n items. Figure 3 illustrates the construction for $k = 3$ and $|U| = 6$.

5 Hierarchical Supervised Schemes

In this section we develop hierarchical schemes that require a fixed amount of work by the supervisor to provide an incentive to workers for doing diligent work, regardless of the total number of workers. As in the previous section, we consider two scenarios: crowdsourcing with and without redundancy.

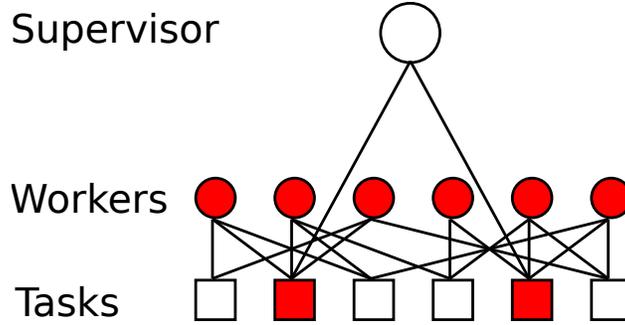


Figure 3: An example of a 3-regular graph where 2 peg tasks (in red) connect all workers.

5.1 Crowdsourcing without Redundancy

The scheme organizes workers into a *supervision tree* (see Definition 1). The internal nodes of the supervision tree represent workers; the leaves represent tasks. A parent-child relation between workers indicates that the child’s loss depends on the parent’s evaluation. A parent node and a child node share one task; this shared item is used to evaluate the quality of the child node’s review work. The root of the tree is the supervisor that provides the truthful supervision.

Definition 1. A supervision tree of depth L is a tree with tasks as leaves, workers as internal nodes, and the supervisor as root. The nodes are grouped into levels $l = 0, \dots, L - 1$, according to their depth; the leaves are the nodes at level $L - 1$ (and are thus all at the same depth). In the tree, workers at level $L - 2$ perform the tasks they are connected to. Every node at level $0 \leq l < L - 2$ performs exactly one task in common with each of its children.

To construct a supervision tree of branching factor at most k , we proceed as follows. We place the tasks as leaves and the above level of workers with at most k tasks per worker. Once level l is built, we build level $l - 1$ by enforcing a branching factor of at most k . For each node x at level l , let y_1, \dots, y_n be its children. For each y_1, \dots, y_n , we pick at random a task s_i performed by y_i , and we assign to x to examine the set $\{s_1, \dots, s_n\}$ of tasks. At the root of the tree, we place the supervisor, following the same method for assigning tasks to review to the supervisor. Figure 4 illustrates a supervision tree with branching factor 2 and depth 3.

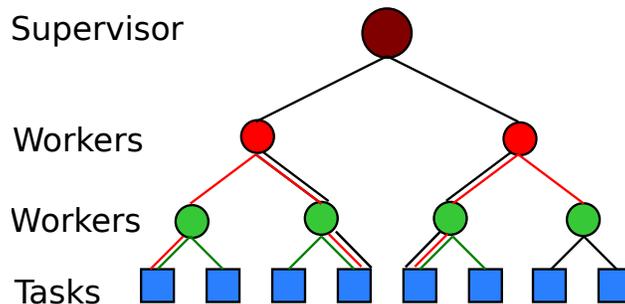


Figure 4: An example of a supervision tree with branching factor 2. The process starts bottom up. Each worker is assigned 2 tasks. For each depth-2 worker, a depth-1 worker is assigned one task in common with worker at depth-2 (red edges). The evaluation of the depth-2 worker will depend on the depth-1 worker. Similarly, the supervisor evaluates a depth-1 worker by reviewing one of the two tasks that the depth-1 worker has done(black edges).

We consider a scenario when each task $i \in I$ has a solution from a set A . The review loss of a workers y in the tree is computed by considering the parent x of y , and the solutions a_x and a_y provided by x and y on the task they both assigned. When the worker y defects, i.e. $a_y \neq a_x$, the loss l_u of the worker is P . When the worker is truthful, the loss is 0.

$$l_u = \begin{cases} P & \text{if } u \text{ defects,} \\ 0 & \text{if } u \text{ is truthful} \end{cases}$$

If the supervisor provides the correct solutions and if the penalty P , which the supervisor can choose, is large enough then under the assumption of rational players, the following theorem proves that when workers are organized into a supervision tree, the truthful strategy is the only Nash equilibrium.

Theorem 3. *If workers are rational and $P > kC$, then the truthful strategy is the only Nash equilibrium of players arranged in a supervision tree with branching factor k .*

Proof. We will prove by induction on the depth $l = 0, 1, \dots, L-1$ of the tree that the only Nash equilibrium for players at depths up to l is the truthful strategy. At depth 0, the supervisor provides the correct solutions, and the result holds trivially, as the supervisor plays a fixed truthful strategy. Let us consider a worker v at depth level k , and denote by I_v the set of tasks assigned by v . We assume that all workers at levels less than k are truthful. Since v does not know which task in I_v has been reviewed also by its parent, and since the parent is truthful, the expected loss l_v of v is

$$l_v = \begin{cases} P/k & \text{if } v \text{ defects,} \\ C & \text{if } v \text{ is truthful} \end{cases}$$

Since the instructor chooses the penalty to be large enough such that $P > kC$, if v is rational and wants to minimize loss, v can only choose the truthful strategy. \square

5.1.1 What information do workers need?

The schemes considered in this section organize workers into hierarchies. What information do workers need to know about the hierarchy, as they set to do their work? Do they need to be given the precise hierarchical scheme, including the names (or identities) of their supervisors? Or can they just be told that a hierarchy exists, without being told even what their place in it is? The interest in these questions lies in the fact that revealing to workers the identity of those above and below them in the hierarchy could create incentives to communicate via secondary channels and sway the outcome.

It turns out that the answer is somewhere in between: while workers do not need to know the identities of the workers above and below them in the hierarchy, they do need to know the level in which they are. The following pair of theorems makes this observation precise.

Theorem 4. *Assume workers are organized into a supervision tree but they are not told their level in the tree. When the number of workers is large enough, defecting is the Nash equilibrium with the lowest loss.*

Proof. Let N and k be the number of players and the tree branching factor respectively. We analyze strategic choices of a worker $u \in U$ when all other workers $U \setminus u$ defect. The probability that the worker u will be reviewed truthfully is $p = k/N$, in all other cases they will be reviewed by the defected workers. The expected loss of worker u when defecting is $l_u = pP$. The loss when u is truthful is $l'_u = C + (1-p)P$. Inequality $l'_u > l_u$ will be satisfied for a large enough k . Indeed, from $l'_u > l_u$ it follows that $\frac{C+P}{2P} > p$. And since $p = k/N$, $N > \frac{2P}{k(C+P)}$. Thus the defecting strategy is a Nash equilibrium. The loss of the equilibrium is pP ; and it can be made arbitrarily small by increasing N .

On the other hand, the truthful strategy is also a Nash equilibrium. When everyone is truthful, it is better to pay cost C and be truthful too, rather than suffer loss P as by our assumption $P > C$. However, the cost C cannot be arbitrarily small. For large enough N , the defecting strategy has the smallest loss. \square

The following theorem essentially says that telling workers their level in the hierarchy is the minimum and sufficient amount of information required to ensure that collaborating is the only Nash equilibrium. Of course, the theorem assumes that the workers can compare notes on the information they are given from the supervisor; otherwise, they could be lied to, and all told that they belong to some arbitrary level in the hierarchy.

Theorem 5. *If there is a fixed upper bound k to the number of tasks that a worker is assigned, then the smallest amount of information a worker needs to know about the hierarchy to have an incentive to play with the truthfully strategy is $\Theta(\log \log N)$, where N is the number of levels in the hierarchy, and $\Theta()$ is the big-Theta notation of complexity theory.*

Proof. If we can give workers $\Theta(\log \log N)$ information or more, then we can tell them their level in the hierarchy, and the above results apply.

Conversely, assume that we can give less than $\Theta(\log \log N)$ bits of information to workers, and consider the situation for $N \rightarrow \infty$. The bits given out would induce a partition C_1, C_2, \dots, C_m of the workers, where workers receiving the same bits would belong to the same class. Assume that the partition classes are sorted according to size, so that $|C_1| < |C_2| < \dots < |C_m|$. As the number of bits is smaller than $\Theta(\log \log N)$, for every $\gamma > 0$, there are n and j so that $|C_j| < \gamma|C_{j+1}|$. In other words, as the number of classes is less than logarithmic in N , as N grows, there must be arbitrarily large gaps in the ratios between sizes of adjacent classes. This implies that, for workers in C_{j+1} as above, the probability of being reviewed by a worker in levels $C_1 \cup \dots \cup C_j$ can become arbitrarily small, since those workers can check on at most $k^2|C_1 \cup \dots \cup C_j|$ workers below them. Thus, defecting becomes the preferred strategy by some of the workers if fewer than $\Theta(\log \log N)$ bits are communicated to the workers. \square

5.1.2 Effects of worker errors in a supervision Tree.

Even though a supervision tree provides an incentive to be truthful, some workers can make mistakes by accident. How does it affect the hierarchy? The following proposition shows that the probability that a truthful worker is treated unfairly does not depend on the tree level, i.e. there is no break down in the hierarchy.

Proposition 3. *Let workers be organized into a supervision tree with branching factor k and let r be a probability that a worker makes a mistake on a tasks and defects. If $P > \frac{kC}{1-2r}$ then the truthful strategy is the only Nash equilibrium and the probability that a truthful worker with level $l > 1$ in the review tree is assigned penalty P is r and does not depend on the level l .*

Proof. The proof that the truthful strategy is the only Nash equilibrium by induction on the tree level and similar to the proof of Theorem 3.

On level $l = 0$ the supervisor plays the truthful strategy. Assume that workers on levels $l = 0, \dots, n - 1$ are truthful by the induction assumption. A worker u on level n minimizes the loss by being truthful. Indeed, if worker u is truthful then their expected loss is $kC + rP$. The kC component is due to cost of performing k tasks and the rP component is due to the probability that the superior on level $n - 1$ deviates and punishes worker u with penalty P . If the worker deviates then their expected loss is $(1 - r)P$. The loss is due to the superior on level $n - 1$ that is truthful with probability $1 - r$. We assume that the loss of the worker u is 0 when both the worker and the superior deviate. The condition that the loss of deviating is greater than being truthful yields inequality $(1 - r)P > kC + rP$. This inequality holds since $P > \frac{kC}{1-2r}$ by the theorem assumption.

We have shown that the truthful strategy is the only Nash equilibrium. Thus a truthful worker is treated unfairly only when their immediate superior accidentally deviates which happens with probability r . \square

5.1.3 Can supervisors be lenient?

We consider a scenario when every superior in a supervision tree evaluates at least two tasks by their subordinate. Can the superior forgive one task and assign a penalty to workers if they deviate in at least 2 tasks? The answer is negative. In fact, if the superior forgives one task then a strategic worker will always deviate on one task. Indeed, when the worker diligently completes all but one of its tasks, the worker can reduce the cost by C by deviating on the last task as the superior assigns a penalty only when the worker deviates on at least 2 tasks.

5.2 Crowdsourcing with Redundancy

In the supervision trees discussed in the previous section, many tasks will have only one worker assigned to it. This is enough to provide an incentive to play the truthful strategy. In some applications, however, it is better to have multiple workers per task. In this section we consider the case of crowdsourcing with redundancy.

We introduce a *supervision hierarchy* that combines a bipartite graph of workers and tasks and a supervision tree. The supervision tree provides an incentive while the bipartite graph ensures that every task is assigned to several workers.

Definition 2. A *supervision hierarchy* is a connected graph that consists of two subgraphs: a bipartite graph $G = (U \cup I, E)$ and a supervision tree T with workers U_T and tasks I_T . The set of tasks I_T is a subset of tasks I and for every worker $u \in U$ there is a task $i \in I_T$ such that the edge (u, i) belongs to E .

Figure 5 illustrates such a supervision hierarchy. The supervisor provide an incentive for the two immediate subordinate workers while these workers provide the incentive to the rest of workers via performing 4 tasks.

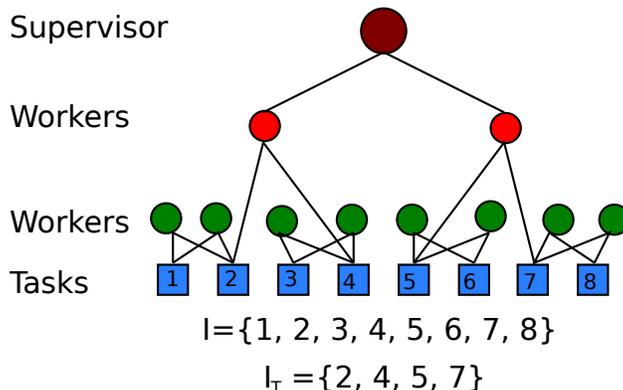


Figure 5: A supervision hierarchy that is a union of a supervision tree and a bipartite graph of workers and tasks. Every task is assigned to at least 2 workers. The set of tasks I_T in the tree is a subset of tasks I in the bipartite graph. Every worker is assigned at least one tasks from the set I_T .

For a given bipartite graph G the task of constructing the smallest supervision hierarchy is \mathcal{NP} -hard. Indeed, the subset I_T of I has the property that every worker $u \in U$ has at least one task from I_T . Thus finding the smallest set I_T is an instance of the *Supervision Assignment* problem we discussed in the previous section; and showed that it is an \mathcal{NP} -hard problem.

We call a supervision hierarchy k -regular if the supervision tree has branching factor k and the bipartite graph is k -regular. The following theorem proves that if $P > kC$, then a k -regular supervision hierarchy provides an incentive for the truthful strategy.

Theorem 6. *If workers are rational and $P > kC$, then the truthful strategy is the only Nash equilibrium of players arranged in a k -regular supervision hierarchy.*

Proof. Let the supervision hierarchy consist of a bipartite graph $G = (U \cup I, E)$ and a supervision tree T with nodes U_T and I_T . According to Theorem 3 the truthful strategy is the only Nash equilibrium in tree T . We will show that the truthful strategy is the only Nash equilibrium in the supervision hierarchy too. Indeed, the loss incurred by workers in the tree T does not depend on workers U , thus, to prove the statement we need to show that when workers U_T provide true solutions, workers U minimize the loss by providing true solutions too. Let a worker $u \in U$ provide a solution a_{iu} on task $i \in I_T$ and let worker $v \in U_T$ provides the correct solution q_i . The loss of worker u is $l_u = l(a_{iu}, q_i)$. If $a_{iu} = q_i$ then the loss is 0, otherwise it is P . The worker u does not know which task is common with a worker from T . In the most favorable scenario to the worker, when the probability of being reviewed is $1/k$, the expected loss of worker u is

$$l_u = \begin{cases} P/k & \text{if } v \text{ defects,} \\ C & \text{if } v \text{ is truthful} \end{cases}$$

Since $P > kC$, the worker u minimizes the loss by playing the truthful strategy. \square

5.2.1 Constructing the Smallest Supervision Hierarchy

Let supervision hierarchy H consist of a bipartite graph $G = (I \cup U, E)$ and a review tree T with leaves $I_T \subseteq I$. A set of workers assigned to task $i \in I$ is ∂i . We denote $\partial' i \subseteq \partial i$ a subset of workers that do not have any tasks from $I_T \setminus i$.

Proposition 4. *Consider a supervision hierarchy H with a bipartite graph G and a review tree T with tasks $I_T \subseteq I$. Assume that for every $i \in I_T$ the set $\partial' i$ has at least 2 workers. If we remove a node v from tree T that is connected to tasks $\{i_1, \dots, i_n\}$ with $n > 1$ and we connect all workers $\partial' i_1, \dots, \partial' i_n$ to tasks from $I_T \setminus i$ to maintain the incentive to be truthful, then the number of edges in H increases.*

Proof. While removing node v from the supervision tree T , we remove $n + 1$ edges: n edges that connect v with tasks $\{i_1, \dots, i_n\}$ and 1 edge that connects v with its superior.

To maintain an incentive to be truthful for workers $\partial' i_1, \dots, \partial' i_n$ that are not connected to tasks from I_T , we connect each of them to a tasks from $I_T \setminus i$. That will add at least $2 * n$ nodes to the supervision hierarchy as each set $\partial' i_1, \dots, \partial' i_n$ contains at least 2 tasks. Therefore the difference between the number of added and removed edges is $2n - n - 1 = n - 1 > 0$. \square

It follows from the Proposition 4 that when we have a control over allocation tasks to workers, we use less edges when we use more peg items rather than requiring all workers review only one task as depicted in figure 2.

6 Quantitative Hierarchical Schemes

In the previous section we considered hierarchical schemes in the binary setting. Workers whether report a correct or incorrect answer in a task. We showed that the truthful strategy is the only Nash equilibrium if the fixed penalty P by the superior is significantly large to outweigh the fixed cost C . In this section we study a quantitative case in which workers can give a real number as an answer in a task.

The results from the previous section carry over to the quantitative case with the difference that reporting a correct answer is replaced with reporting the correct answer with a certain variance v . The magnitude of the variance v is controlled by the appropriate loss function of the incentive mechanism.

In detail, consider a set U of workers; these workers can all operate in one level, or they can be organized into a hierarchy. The workers are asked to evaluate items from a set I ; every item $i \in I$ has an intrinsic quality $q_i \in \mathbb{R}$. In order to produce a measurement of the quality of an item to within variance v , a worker needs to pay a price $f(v)$, where f is a non-negative, monotonically decreasing function defined on the set \mathbb{R}^+ of strictly positive variances. Note that the hypothesis that f is monotonically decreasing is not restrictive: if $f(v) < f(v')$ for $v < v'$, then a user could simply buy a measurement of variance v for the smaller price $f(v)$ and then add noise with variance $v' - v > 0$, rather than pay the higher price $f(v')$.

In order to produce an incentive towards precise work, some of the evaluations performed by each user are evaluated also by someone higher up in the hierarchy, as in the previous sections. If the user produces estimate x , while the supervisor produces estimate y , the user is penalized using the loss function

$$\mathcal{L}(x, y) = \alpha(x - y)^2$$

where $\alpha > 0$ is a penalty constant. If the user's estimate has variance v and the supervisor's estimate has variance w , if the user and supervisor estimates are uncorrelated (which we expect to be the case), by the law of additivity of variances, the expected penalty of a user is $p\alpha(w + v)$, where $p \in [0, 1]$ is the probability that an item rated by the user is also rated by a supervisor. Thus, the optimal behavior for the user consists in using a variance v that minimizes the overall cost

$$p\alpha(w + v) + f(v) = p\alpha w + p\alpha v + f(v) . \quad (2)$$

Thus, the optimal behavior for the user consists in using a variance that is a global minimum of $p\alpha v + f(v)$. We note that such global minima do not depend on w : hence, the precision of estimates in a hierarchical scheme is constant for each level, and does not degrade as the depth in the hierarchical revision tree increases.

Lemma 1. *The set M of global minima of the cost function $c(v) = p\alpha(w + v) + f(v)$ has a maximum value $v^+ \in \mathbb{R}$, independent of w .*

The lemma follows from the fact that f is non-negative, so that $c(v) < c(v')$ for all $v' > v + f(v)/p\alpha$. The following theorem makes precise the fact that the optimal strategies for a user involve a variance that does not depend on the level of the user in the revision hierarchy.

Theorem 7. *If the cost $f(v)$ for a worker to achieve variance v is non-negative, and if the loss function for disagreeing with a supervisor is as indicated in (2), then for each $\alpha > 0$ and $p \in [0, 1]$, there is $v^+ \in \mathbb{R}$ such that it is optimal for a worker to use a variance $v < v^+$, regardless of the precision w of the worker's supervisor.*

Intuitively, the theorem depends on the fact that the loss function (2) is additive in the variance of the worker, so that any extra increase in worker variance corresponds directly to an increase in worker loss, independently from the precision of the worker's supervisor. As a consequence of the theorem, the precision of workers that play optimally according to the incentives does not depend on their level in the hierarchy, nor on their belief on the accuracy of their supervisors.

7 Conclusions

We proposed and analyzed supervising schemes that lead to high-quality crowdsourced input by imposing truthful incentives to workers. The supervisor of a crowdsourcing system can use our schemes to organize the crowd in layers, maintaining the cost of supervision very low irrespective of the total size of the crowd which can be arbitrarily large. In a crowd of well-performing workers organized with our mechanism, individual workers serve their interest best by also providing truthful answers to their assigned tasks. Moreover,

there exist no other equilibria, ensuring that workers will not evade work by colluding to game the system. The scheme is easy to implement as the workers perform the same type of tasks at any depth in the hierarchy, that is, there are no special meta-review tasks. Workers only need to be informed about their depth in the hierarchy to maintain their incentives, without any requirement for additional skills or instruction. Our schemes graciously extend from simple binary verifiable tasks to quantitative tasks making them relevant to a wide range of crowdsourcing applications.

References

- [1] N. Ailon. Aggregation of partial rankings, p-ratings and top-m lists. *Algorithmica*, 57(2):284–300, 2010.
- [2] N. Alon, F. Fischer, A. Procaccia, and M. Tennenholtz. Sum of us: Strategyproof selection from the selectors. In *Proceedings of the 13th Conference on Theoretical Aspects of Rationality and Knowledge, TARK XIII*, pages 101–110, New York, NY, USA, 2011. ACM.
- [3] A. Carvalho, S. Dimitrov, and K. Larson. Inducing honest reporting without observing outcomes: An application to the peer-review process. *arXiv preprint arXiv:1309.3197*, 2013.
- [4] R. T. Clemen. Incentive contracts and strictly proper scoring rules. *Test*, 11(1):167–189, 2002.
- [5] A. Dasgupta and A. Ghosh. Crowdsourced judgement elicitation with endogenous proficiency. In *Proceedings of the 22nd international conference on World Wide Web*, pages 319–330. International World Wide Web Conferences Steering Committee, 2013.
- [6] L. de Alfaro and M. Shavlovsky. Crowdgrader: a tool for crowdsourcing the evaluation of homework assignments. In *The 45th ACM Technical Symposium on Computer Science Education, SIGCSE '14, Atlanta, GA, USA - March 05 - 08, 2014*, pages 415–420, 2014.
- [7] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th international conference on World Wide Web*, pages 613–622. ACM, 2001.
- [8] A. Ghosh. Game theory and incentives in human computation systems. In *Handbook of Human Computation*, pages 725–742. Springer, 2013.
- [9] S. Johnson, J. W. Pratt, and R. J. Zeckhauser. Efficiency despite mutually payoff-relevant private information: The finite case. *Econometrica: Journal of the Econometric Society*, pages 873–900, 1990.
- [10] R. Jurca and B. Faltings. Enforcing truthful strategies in incentive compatible reputation mechanisms. In *Internet and Network Economics*, pages 268–277. Springer, 2005.
- [11] R. Jurca and B. Faltings. Minimum payments that reward honest reputation feedback. In *Proceedings of the 7th ACM conference on Electronic commerce*, pages 190–199. ACM, 2006.
- [12] R. Jurca and B. Faltings. Mechanisms for making crowds truthful. *Journal of Artificial Intelligence Research*, 34(1):209, 2009.
- [13] E. Kamar and E. 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] R. M. Karp. *Reducibility among combinatorial problems*. Springer, 1972.
- [15] D. Kurokawa, O. Lev, J. Morgenstern, and A. D. Procaccia. Impartial peer review. *To be submitted*.
- [16] N. Miller, P. Resnick, and R. Zeckhauser. Eliciting informative feedback: The peer-prediction method. *Management Science*, 51(9):1359–1373, 2005.
- [17] M. J. Osborne and A. Rubinstein. *A course in game theory*. MIT press, 1994.
- [18] D. Prelec. A bayesian truth serum for subjective data. *science*, 306(5695):462–466, 2004.
- [19] K. Raman and T. Joachims. Methods for ordinal peer grading. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 1037–1046, New York, NY, USA, 2014. ACM.
- [20] T. Walsh. The peerrank method for peer assessment. *CoRR*, abs/1405.7192, 2014.
- [21] R. L. Winkler and A. H. Murphy. “Good” probability assessors. *Journal of applied Meteorology*, 7(5):751–758, 1968.
- [22] C. Godsil and G. F. Royle. *Algebraic graph theory*. Springer Science & Business Media, 207, 2013
- [23] E. Halperin. Improved approximation algorithms for the vertex cover problem in graphs and hypergraphs. *SIAM Journal on Computing*, 31(5):1608–1623, 2002
- [24] B. Waggoner and Y. Chen. Output agreement mechanisms and common knowledge. Second AAAI Conference on Human Computation and Crowdsourcing, 2014
- [25] J. Witkowski, Y. Bachrach, P. Key, and D. C. Parkes. Dwelling on the negative: Incentivizing effort in peer prediction. First AAAI Conference on Human Computation and Crowdsourcing, 2013
- [26] B. Faltings, J. J. Li, and R. Jurca. Eliciting truthful measurements from a community of sensors. Internet of Things (IOT), 2012 3rd International Conference on the, pages 47–54, 2012. IEEE.
- [27] J. Witkowski and D. C. Parkes. A Robust Bayesian Truth Serum for Small Populations. AAAI, 2012
- [28] Y. Cai, C. Daskalakis and C. Papadimitriou. Optimum statistical estimation with strategic data sources. ArXiv e-prints, 2014
- [29] G. Radanovic and B. Faltings. A robust bayesian truth serum for non-binary signals. Proceedings of the 27th AAAI Conference on Artificial Intelligence, AAAI 2013, EPFL-CONF-197486:833–839, 2013
- [30] G. Radanovic and B. Faltings. Incentives for truthful information elicitation of continuous signals. Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014
- [31] B. Riley. Minimum truth serums with optional predictions. Proceedings of the 4th Workshop on Social Computing and User Generated Content (SC14), 2014
- [32] P. Zhang and Y. Chen. Elicitability and knowledge-free elicitation with peer prediction. Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems, pages 245–252, 2014
- [33] C. Piech, J. Huang, Z. Chen, C. Do, Chuong, A. Ng, and D. Koller. Tuned models of peer assessment in MOOCs. arXiv preprint arXiv:1307.2579, 2013

- [34] V. Kamble, N. Shah, D. Marn, A. Parekh, and K. Ramachandran. Truth Serums for Massively Crowdsourced Evaluation Tasks. arXiv preprint arXiv:1507.07045, 2015
- [35] Y. Kong, G. Schoenebeck, and K. Ligett. Putting Peer Prediction Under the Micro (economic) scope and Making Truth-telling Focal. arXiv preprint arXiv:1603.07319, 2016
- [36] P. Venetis and H. Garcia-Molina. Quality control for comparison microtasks. Proceedings of the first international workshop on crowdsourcing and data mining, pages 15–21. ACM.
- [37] C. Harris. Youre hired! an examination of crowdsourcing incentive models in human resource tasks. Proceedings of the Workshop on Crowdsourcing for Search and Data Mining (CSDM) at the Fourth ACM International Conference on Web Search and Data Mining (WSDM), pages 15–18, 2011
- [38] D. Oleson, A. Sorokin, G. P. Laughlin, V. Hester, J. Le, and L. Biewald. Programmatic Gold: Targeted and Scalable Quality Assurance in Crowdsourcing. Human computation, 11(11), 2011