Incentives and Truthful Reporting in Consensus-centric Crowdsourcing

Ece Kamar, Eric Horvitz
{eckamar,horvitz}@microsoft.com

February 2012

Technical Report
MSR-TR-2012-16

We address the challenge in crowdsourcing systems of incentivizing people to contribute to the best of their abilities. We focus on the class of crowdsourcing tasks where contributions are provided in pursuit of a single correct answer. This class includes citizen science efforts that seek input from people with identifying events and states in the world. We introduce a new payment rule, called consensus prediction rule, which uses the consensus of other workers to evaluate the report of a worker. We compare this rule to another payment rule which is an adaptation of the peer prediction rule introduced by Miller, Resnick and Zeckhauser to the domain of crowdsourcing. We show that while both rules promote truthful reporting, the consensus prediction rule has better fairness properties. We present analytical and empirical studies of the behavior of these rules on a noisy, real-world scenario where common knowledge assumptions do not necessarily hold.

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
http://www.research.microsoft.com

1 Introduction

Over the last five years, Web-based crowdsourcing platforms have become available for providing programmatic access to populations of workers with diverse capabilities. Crowdsourcing systems include *games with a purpose* [21] and *task markets* such as Mechanical Turk. We address a central challenge with the harnessing of human talent via these systems: incentivizing workers to contribute to the best of their abilities.

Designing a crowdsourcing application involves the specification of incentives for services and the checking of the quality of contributions. Methodologies for checking quality include providing a payment if the work is approved by the task owner and also hiring additional workers to evaluate contributors' work. Both of these approaches place a burden on people and organizations commissioning tasks. And there are multiple sources of inefficiency: Recent experiments on Mechanical Turk have demonstrated that task markets may be negatively affected by the strategic behaviors of workers and task owners [12, 15]. For example, there can be strategic manipulation of work by participants that reduces their contribution but increases payments. And task owners may prefer to reject contributions simply to reduce the payments they owe to the system. Moreover neither a task owner nor the task market may know the task well enough to be able to evaluate worker reports. In this paper, we introduce incentive mechanisms that promote truthful reporting among workers of a crowdsourcing system and prevent task owner manipulations.

We focus on a specific class of crowdsourcing tasks that we refer to as *consensus tasks*, but the ideas presented here can be generalized to many settings in which multiple reports collected from people are used to make decisions. Consensus tasks are aimed at determining a single correct answer or a set of correct answers to a question or challenge, such as identifying labels for items, quantities, or events in the world, based on multiple noisy reports collected from human workers. Consensus tasks include a large swath of *citizen science* projects where non-experts assist with identifying ground truth. An example of a large-scale consensus task is the classification of galaxies identified in a comprehensive sky survey, where workers label galaxies based on their appearance in astronomical images [13]. Consensus tasks can also be subtasks of a larger complementary computing task, where a computer system is recruiting human workers to solve pieces of a larger problem that it cannot solve. For example, a computer system for providing real-time traffic directions may recruit drivers from a certain area to report about traffic conditions, so that the system is able to provide up-to-date directions more confidently.

We study different payment rules for incentivizing workers in crowdsourcing systems and the properties of these rules. We first analyze existing payment rules used in consensus tasks and show that they are vulnerable to worker manipulations. Next, we adapt *peer prediction rules* to the domain of crowdsourcing. These rules were introduced by Miller, Resnick and Zeckhauser to gather honest reviews of products by making use of proper scoring rules [16]. In our adaptation to the domain of crowdsourcing, these rules pay a worker depending on how

well her report helps to predict another worker's report for the same task. We demonstrate the way payments can be computed with the peer prediction rules for consensus tasks based on the other components of a system, and we study the important challenges in applying them in real-world multi-agent systems.

To address some shortcomings of the peer prediction rule, we introduce a novel payment rule, called *consensus prediction rule*. This more sophisticated payment rule couples payment computations with planning to generate a robust signal for evaluating worker reports. This rule rewards a worker based on how well her report can predict the consensus of other workers. It incentivizes truthful reporting, while providing better fairness than peer prediction rules.

Peer prediction and consensus prediction rules make strong common knowledge assumptions to promote truthful reporting. For the domain of consensus tasks, these assumptions mean that every worker shares the same prior about the likelihoods of answers and the likelihoods of worker reports, and the system knows this prior. This assumption is one of the biggest obstacles in applying peer and consensus prediction rules in a real-world system, in which these likelihoods can only be predicted based on noisy predictive models. We study approaches for relaxing this assumption. We show that in settings where common knowledge assumptions do not hold, workers can be incentivized to communicate and collaborate with the system to correctly estimate the true prior, and the true likelihoods of worker reports.

We empirically evaluate different payment rules with simulations which validate their truthfulness and fairness properties. Our analyses show that crowdsourcing systems that are not implementing incentive compatible payments may suffer significantly, and that payments calculated with the consensus prediction rule achieve high levels of fairness. To the best of our knowledge, we present the first empirical study of the behavior of peer prediction and consensus prediction rules with strategic agents when common knowledge assumptions do not hold. We study settings in which payments are computed based on predictions of noisy models. The study shows that the incentive compatibility properties of these rules are robust to noise.

2 Related Work

With the recent advances in using crowdsourcing for solving tasks that computers cannot easily do alone, there has been growing interest in principles and algorithms for efficient crowdsourcing. An active area of research has been on understanding and modeling worker behavior in task markets, including efforts for predicting reservation wage for a given task [8], for predicting the way task features and payment features affect the quality of outcome [9], for predicting worker contribution as a function of payment [14], and for identifying content contributors that are untrustworthy or inaccurate [5]. Another line of research has focused on planning algorithms to determine the optimal task structure and hiring policies for solving a task with crowdsourcing [19, 3, 11, 20].

In contrast to efforts to address challenges in modeling and planning, there

has been little work on building incentive mechanisms for crowdsourcing systems. Find-fix-verify has been proposed as a policy that hires additional workers for quality tracking [1]. Another approach suggests verifying the competency and honesty of workers by asking a set of easily verifiable questions [12, 19]. Both of these approaches introduce additional costs either in the form of recruiting additional workers or burdening workers with redundant questions. There has been complementary work on incentivizing workers of a crowdsourcing system to attract competent peers to contribute to a task [23, 4].

Prediction markets and market scoring rules incentivize truthful reporting for tasks for which the true answer is eventually revealed [2, 7]. However for many crowdsourcing tasks the true answer may never be revealed and such tasks are the focus of this paper. This paper builds on the prior work on peer-prediction methods to promote truthful reporting without inducing additional costs. Peer-prediction methods were first introduced by [16] to gather honest reviews of products, papers, and proposals by making use of proper scoring rules. These methods are supported by empirical evidence showing that people indeed learn to maximize their payoff by reporting truthfully when payments are calculated with respect to a proper scoring rule [17]. *Collective revelation* is an alternative approach that inherently weighs an agent's report more if the agent is more informative, but it requires each agent to report twice, based on a consideration with and without evidence, and is only valid if the distribution over the correct outcome has a well-understood structure (e.g., Normal distribution) [6]. The *Bayesian Truth Serum* makes weaker assumptions on common knowledge, but requires each agent to report predictions about other agents' predictions, which may be too costly for workers within a crowdsourcing system [18].

3 A Mechanism for Solving Consensus Tasks

Our overarching goal is to design autonomous systems that collect information from people to recover the true state of the world, and thus to make effective decisions. We focus on crowdsourcing systems for solving consensus tasks as a real-world example.

An automated system for consensus tasks has access to a population of workers, who are able to make inferences about the correct answer of a consensus task. We use the term "worker's inference" to refer to the worker's true belief about the correct answer of a task. A worker's report to the system may differ from the inference, for example if the worker strategizes about what to report. The goal of the system is to deduce an accurate prediction of the correct answer of a task by making use of multiple worker reports. We now formally define consensus tasks, and then present a system design for solving these tasks.

Definition 1. *Let I denote the set of workers in worker population, $A = \{a_1, ..., a_n\}$ denote the set of possible answers for task $t \in T$. f is the set of features describing the task and workers. Task t is a consensus task if,*

- *There exists a mapping $t \rightarrow a^* \in A$, where a^* is the correct answer of task t.*
- *Let A^* be a random variable for the correct answer of a given task, and C_p be another random variable for the answer inferred by a random worker in the population. A^* is stochastically relevant for C_p conditional on f. That is, for any distinct realization of A^*, $\tilde{a}$ and $\bar{a}$, there exists a realization of C_p, c_p, such that*
  $Pr(C_p = c_p | A^* = \tilde{a}, f) \neq Pr(C_p = c_p | A^* = \bar{a}, f)$.
- *Let C_i be a random variable denoting the answer inferred by worker i, and C_j be another variable denoting the answer inferred by a random worker from the remaining population $I_{-i} = I \setminus \{i\}$. For any worker i in the worker population, C_i is stochastically relevant for C_j conditional on f.*

For simplicity, Definition 1 assumes consensus tasks to have a single correct answer; however, the results presented in this work generalize to cases in which a set of answers may serve as correct answers. The second condition of Definition 1 ensures that the worker population is informative for a given task. The third condition is the foundation of the truth promoting payment rules that we will focus on later. This condition is realistic for many domains in which worker inferences about a task depends on the correct answer of the task or the hidden properties of the task, thus a worker's inference helps to predict other workers' inferences. For example, a worker of the Galaxy Zoo system classifying a galaxy as a spiral galaxy increases the probability that another worker will provide the same classification [13].

A successful crowdsourcing system needs to satisfy both task owners and workers. Thus, the system designers face two key challenges: (1) generating a policy for solving a given task, and (2) providing compelling and fair incentives to workers. To address these challenges, a system for solving consensus tasks needs to generate models that predict the correct answer of a task at any point during execution as well as the worker reports that will be obtained by the system. In addition, based on these models, the system needs a policy for deciding whether to hire a new worker or to terminate and deliver the most likely answer to the task owner, and provide payments to workers in return for their effort. The detailed investigations of learning these predictive models and developing policies for consensus tasks have been presented separately [11]. In this work, we provide a summary of the key findings from this previous work, and focus on the unanswered challenge of designing incentives for workers.

The models for predicting the correct answer and for predicting worker reports makes inferences based on a set of features that represent the characteristics of tasks and workers. To build these models, the system collects data about the system, workers, and tasks being executed. For a given task, feature set $\mathcal{F}_t$ include features that are initially available in the system. $\mathcal{F}_t$ may contain features of the task (e.g., task difficulty, task type and topic), features of the general worker population (e.g., population competency), and features about the components of the system (e.g., minimum and maximum incentives

offered). Feature set $\mathcal{F}_{w_i}$ includes features of a particular worker i, which may include the personal competency of the worker, her availability and her abilities. Feature set $\mathcal{F}_i = \mathcal{F}_{w_i} \cup \mathcal{F}_t$ represents the complete set of evidential observations or features relevant for making predictions about worker i's report. After collecting m worker reports, $\mathcal{F} = \mathcal{F}_t \cup \mathcal{F}_{w_1} \cup ... \cup \mathcal{F}_{w_m}$ represents the complete set of evidential observations or features relevant for predicting the correct answer of a task. $\mathcal{F}$ may contain hidden features (e.g., the difficulty of a task), which may need to be predicted to make accurate inferences about the correct answer and about the worker reports. $\mathcal{F}_i$ is provided as input to the model that predicts the report of worker i. The full feature set $\mathcal{F}$ is provided as input to the model that predicts the correct answer of a task. For simplicity of notation, $Pr(X|F = f)$ denoted as $Pr_f(X)$ throughout the paper.

The system harnesses two predictive models for making hiring decisions and for calculating payments: The answer model (M_A) and the report model (M_R) (See [11] for details on these models). $M_A(a, f_t)$ is the prior probability of the correct answer being a given the initial feature set of the task. For example, if a galaxy has features that resemble a spiral, the prior probability of this galaxy being a spiral galaxy is higher. $M_R(r_i, a^*, f_i)$ is the probability of worker i reporting r_i given that the correct answer of the task is a^* and the set of features relevant to the worker report is f_i. The likelihood of a worker identifying a galaxy correctly may depend on the features of the task and of the worker. This likelihood tends to be relatively higher if the galaxy is easy to classify, or the worker is competent. Since $\mathcal{F}_k$ includes all relevant features to predict any k^{th} worker's report, for all worker couples i and j, R_i and R_j are independent given $\mathcal{F}_i$, $\mathcal{F}_j$ and A^*.

At each point during execution, the system makes a decision about whether to hire a new worker or terminate the task. When it decides not to hire additional workers, it deducts a consensus answer $\hat{a}$ based on aggregated worker reports and delivers this answer to the owner of the task. Given a sequence of reports collected from workers, $r = \{r_1, ..., r_m\}$, it chooses $\hat{a}$ as given below:

$$\hat{a} = \underset{a \in A}{\operatorname{argmax}} Pr_f(A^* = a | R_1 = r_1, ..., R_m = r_m)$$

The system implements a policy for deciding when to stop hiring workers and deliver the consensus answer to the task owner. For simplicity of analysis, we limit policies to make decisions about how many workers to hire and not to make decisions about who to hire and how much to pay. A sample policy that we will be using through the paper continuously checks whether the system's confidence about the correct answer has reached a threshold value T. The policy hires a new worker if target confidence T has not been reached after receiving a sequence of reports r:

$$(\max_{a \in A} Pr(A^* = a | R = r, \mathcal{F} = f)) < T$$

A more sophisticated policy that can make hiring decisions by solving a Partially Observable MDP has been introduced in previous work [11].

Let π be the policy implemented by the system. We define a function M_π such that for a given sequence of worker reports r and feature set f, $M_\pi(r, f)$ is $\emptyset$ if π does not terminate after receiving r, and is $\hat{a}$, the consensus answer, otherwise.

4 Incentives for Truthful Reporting

Among various factors that motivate workers, including enjoyment, altruism and social reward, monetary payments are the most generalizable and straightforward to replicate. We shall focus on quantifiable payments as incentives in crowdsourcing tasks, which can be monetary payments or reputation points.

Following the literature on prediction markets, an intuitive approach to rewarding workers in consensus tasks is rewarding agreements with the correct answer. A challenge with this approach is that the correct answer may take too long to be revealed—or may never be revealed. Moreover, the signal about the correct answer may be unreliable; if the correct answer is revealed by the task owner, the owner may have an incentive to lie to decrease payments. We now present payment rules that reward workers without knowing the correct answer. These rules use peer workers' reports to evaluate a worker, and does not require input from task owners, thus prevents task owner manipulations.

4.1 Preliminary

In this section, we present the background, definitions and analysis that are needed to formalize payment rules for consensus tasks. We start by stating our assumptions in designing payments. In consensus tasks, workers report on a task once and maximize their individual utilities for the current task. We follow the common knowledge assumptions made by the prior work on peer prediction methods. These common knowledge assumptions translate to the domain of consensus tasks as follows: The probability assessments performed by models M_A and M_R are accurate and common knowledge. These assumptions can be realized by a crowdsourcing system by collecting evidence about previous tasks and workers, and by building accurate predictive models. For cases in which predictions of the system are accurate but individual workers' predictions are not, the assessments of the system can be made common knowledge with public revelation. In Section 5, we explore approaches for relaxing common-knowledge assumptions in real-world systems.

We model a consensus task as a game of incomplete information in which players' strategies consist of their potential reports [1]. We perform Bayesian-Nash equilibrium analysis to study the properties of payment rules. A worker's report is evaluated based on a peer worker's report for the same task or a subset of such reports. $\tau_i(r_i, r_{-i}) \rightarrow \bar{\mathbb{R}}$ denotes the system's payment to worker i, based on r_i, worker i's report, and r_{-i}, a sequence of reports collected for the same

[1] Our analysis focuses on the reporting behavior of workers once they decide to participate in a consensus task.

task excluding r_i. C_{-i} is a random variable for the sequence of inferences by all workers except worker i. Ω_R is the domain of worker inferences and reports. Let s_i^t be a reporting strategy of worker i such that for all possible inferences c_i she can make for task t, $s_i^t(c_i \in \Omega_R) \rightarrow r_i \in \Omega_R$. s^t is a vector of reporting strategies for all workers reporting to the system, s_{-i}^t is defined as $s^t \setminus \{s_i^t\}$.

s^t is a strict Bayesian-Nash equilibrium of the consensus task t if, for each worker i and inference c_i,

$$\sum_{c_{-i}} \tau_i(s_i^t(c_i), s_{-i}^t(c_{-i}))\ Pr_f(C_{-i} = c_{-i}|C_i = c_i) > \sum_{c_{-i}} \tau_i(\hat{r}_i, s_{-i}^t(c_{-i}))\ Pr_f(C_{-i} = c_{-i}|C_i = c_i)$$

for all $\hat{r}_i \in \Omega_R \setminus \{s_i^t(c_i)\}$.

A strategy s_i^t is truth-revealing if for all $c_i \in \Omega_R$, $s_i^t(c_i) = c_i$. $\mathcal{M} = (t, \pi, \tau)$, mechanism for task t with policy π and payment rule τ, is strict Bayesian-Nash incentive compatible if truth-revelation is a strict Bayesian-Nash equilibrium of the task setting induced by the mechanism[2].

We use proper scoring rules as the main building blocks for designing payment rules that promote truthfulness in consensus systems. We define proper scoring rules for the forecast of a categorical random variable. The set of possible outcomes for the variable is $\Omega = \{\omega_1, ..., \omega_n\}$. A forecaster reports a forecast p, where p is a probability vector $(p_1, ..., p_n)$, and p_k is the probability forecast for outcome ω_k. A proper scoring rule S takes as input the probability vector p and the realized outcome of the variable ω_i, and outputs a reward in $\bar{\mathbb{R}}$ for the forecast. Let the probability vector q be the forecaster's true forecast for the random variable, a function S is a strictly proper scoring rule if the expected reward is maximized when $p = q$. Function S measures the performance of a forecast in predicting the outcome of a random variable. Three well-known strictly proper scoring rules are:

1. Logarithmic scoring rule:

$$S(p, \omega_i) = ln(p_i)$$

2. Quadratic scoring rule:

$$S(p, \omega_i) = 2p_i - \sum_{\omega_k} p_k$$

3. Spherical scoring rule:

$$S(p, \omega_i) = \frac{p_i}{(\sum_{\omega_k} p_k^2)^{1/2}}$$

Next, we present the general idea for using proper scoring for calculation of truth-promoting payments in consensus tasks. We pick a public signal for

[2]Task t specifies the strategies available for workers as well as features of the task and the worker population.

which a worker's report is stochastically relevant for. The worker's report gives a clue about what the value of the signal will be. We use the worker's report to generate a forecast about the signal and reward the worker based on how well the forecast predicts the realized value of the signal. From the definition of proper scoring rules, the reward of the worker is maximized when $r_i = c_i$. In the following sections, we propose signals that can be used to evaluate worker reports and provide methods for calculating the payment of a worker reporting to a real-world consensus system.

4.2 Applying Existing Payment Rules To Consensus Tasks

We now explore different payment rules that have been proposed by previous work within and beyond the literature on crowdsourcing systems. We describe how these rules can be computed for consensus tasks and analyze their incentive compatibility and fairness properties.

4.2.1 Basic Payment Rules

A number of crowdsourcing systems have implemented payment rules that reward workers based on agreement among workers. Two examples of these rules are the *basic peer rule* which rewards a worker if her report agrees with a randomly selected worker's report on the same task (e.g., implemented in the ESP game [21]) and the *basic answer rule* which rewards a worker if her report agrees with the consensus answer (e.g., some tasks in Mechanical Turk). We refer to these rules as *basic payment rules* as worker payments depend on agreements among the reports of workers, independent of the likelihood of agreement.

Basic payment rules are not guaranteed to promote truthful reporting for consensus tasks. We propose an example from the Galaxy Zoo domain to demonstrate that systems implementing basic payments might be negatively affected by strategic reporting. We will use this example continuously in this paper for demonstrating properties of other payment rules. For the example, the correct classification of any galaxy can either be elliptical (e) or spiral (s). Based on the features of a given galaxy, the priors for the type of the galaxy are $Pr(A^* = e) = 0.8$ and $Pr(A^* = s) = 0.2$. The accuracy of each worker is 70% in predicting the correct answer. The consensus system implements a simple policy that terminates after hiring 4 workers. For all basic payment rules, the best response of a worker is always reporting e, when other workers are reporting truthfully, or reporting strategically. Thus, it is not possible to solve this task with strategic workers if basic payments are provided as incentives.

The absence of incentive compatibility in a system introduces important challenges for both workers and the system. To maximize payments, workers need to strategize and need to reason about other workers and the likelihood of their reports, which may be a difficult cognitive challenge. Since the worker reports are not guaranteed to be truthful, the policy implemented in the mechanism needs to reason about the different strategies that workers may employ, which makes the planning process harder.

4.2.2 Peer Prediction Rule

The *peer prediction rule* is proposed by Miller et. al. for domains in which users' truthful reviews about products and services are valuable. This rule assumes that a user's review about a product is stochastically relevant for another user's review about the same product. A user is evaluated based on how well her review helps to predict the other user's review, and a proper scoring rule is used to calculate the reward of the user for writing the review. In this section, we show that the peer prediction is a natural rule to incentivize workers of a consensus system. We reward a worker based on how well her report can predict the report of another worker. In addition to demonstrating that this payment rule promotes truthful reporting in consensus systems, we investigate multiple issues that have not been investigated before in applying this rule to a real-world system.

Proposition 1. *For a given consensus task t and policy π, let r_j be the report of a random worker from I_{-i}. $M = (t, \pi, \tau^p)$ is strict Bayesian-Nash incentive compatible, where worker i's payment, τ_i^p, for reporting to task t is,*

$$\begin{aligned} \tau_i^p(r_i, r_j) &= S(p^p, r_j), \text{ where} \\ \text{for all } r_k \in \Omega_R,\ p_k^p &= Pr_f(C_j = r_k | C_i = r_i) \end{aligned}$$

Proof. Given the definition of consensus tasks and the fact that C_i is stochastically relevant for C_j given f, the proof follows from the definition of proper scoring rules. □

As long as a worker i trusts the system to accurately calculate $Pr_f(C_j|C_i = r_i)$ and believes that other workers report honestly, it is a best response for the worker to report truthfully without performing any complex calculations. For the Galaxy Zoo example presented earlier, when other workers are reporting truthfully, a worker inferring the correct answer of a galaxy as s has a higher expected payment for reporting truthfully than reporting the likely label e.

Calculations of conditional probabilities needed for computing these payments is a central problem that is not addressed by previous work. In a consensus system, the probability distribution $Pr_f(C_j|C_i)$ may depend on the identity of workers reporting to the system as well as the features of the task they are reporting for. For example, this conditional probability is different if one of the workers is highly competent or the task is difficult. In the equilibrium when all workers report their true inference, we show below that $Pr_f(C_j|C_i)$ can be computed by applying the Bayes rule and by making use of answer and report models presented in Section 3 based on the set of features f.

$$Pr_f(C_j = r_j | C_i = r_i) = \frac{\sum_{a \in A} M_A(a, f_t) M_R(r_i, a, f_i) M_R(r_j, a, f_j)}{\sum_{a \in A} M_A(a, f_t) M_R(r_i, a, f_i)}$$

A direct enhancement of the peer prediction rule is the *average peer prediction rule*, τ^a. For any worker i contributing to r, τ_i^a is computed as below.

$$\tau_i^a(r_i, r_{-i}) = \sum_{r_j \in r_{-i}} \frac{\tau_i^p(r_i, r_j)}{|r_{-i}|}$$

The incentive compatibility property of peer prediction rule holds for τ^a.

We revisit the example given in Section 4.2.1 to analyze the fairness properties of peer prediction rules. It is possible to normalize the peer prediction payments calculated with the quadratic or spherical proper scoring rules to [0,1] interval (or any desired interval) without impairing their incentive compatibility properties by calculating the minimum and maximum payments that can be received for task (i.e., a linear transformation of a proper scoring rule is a proper scoring rule). A galaxy task has the correct answer e and receives the sequence of reports $\{e, s, e, e\}$. The sets of normalized payments computed by τ^p and τ^a are $\{0.66, 0, 1, 1\}$ and $\{0.89, 0, 0.89, 0.89\}$. As shown by this example, the payments computed by the peer prediction rule are affected by the randomness in selecting the worker for comparison. Two workers of the same competency predict the answer correctly, but receive different payments. Moreover, both sets of payments suffer from the variance in worker reports. A worker reporting correctly receives a lesser payment because there is another worker reporting incorrectly. Thus, this worker may envy the higher amount of payment another competent worker may receive for reporting to a task for which every worker correctly reports.

Fairness is important for the happiness of workers and health of the system as fair incentives reward successful workers and motivate them to participate and do their best in the system. As shown by the analysis above, the sensitivity of peer prediction payments to variance in worker reports result in diminished fairness for workers. In the next section, we propose a more sophisticated payment rule that provides higher levels of fairness to workers of a consensus system.

4.3 Consensus Prediction Rule

We now present a novel payment rule, called the *consensus prediction rule*, which rewards a worker according to how well her report can predict the outcome of the system (i.e., the consensus answer that will be decided by the system), if she was not participating in it. Calculation of this payment for the worker is a two-step process. In the first step, we use the worker's report as a new feature to update the system's predictions about the likelihood of answers and worker reports. Based on these updated predictions, we simulate the system to generate a forecast about the likelihoods of possible consensus answers. In the second step, we use reports from all other workers to predict the most likely consensus answer as if the worker in question never existed. The worker is rewarded based on how well the forecast generated based on only her report can predict the realized consensus answer by her peers. This payment rule forms a direct link between a worker's payment and the outcome of this system. Because

the outcome of a successful system is more robust to erroneous reports than the signal used in peer prediction rules, this payment rule has better fairness properties.

Before we go into the formal definition of the consensus prediction rule, and its calculations, we demonstrate the way this rule is computed on our Galaxy Zoo toy example. In this example, the system follows the simple policy that terminates after collecting reports from four workers. Let's assume that we collect report sequence $\{e, s, e, e\}$. As an example, we calculate the payment for the first worker. This worker reporting e increases the likelihood of the correct answer being e and other workers reporting e. To generate the forecast about the consensus answer, as if we do not have access to any real worker reports, we simulate all possible report sequences from four hypothetical workers. Next, we calculate the likelihood of each simulated sequence, along with the consensus answer for that sequence, based on updated answer priors and report likelihoods. The cumulative likelihoods of consensus answers over all possible report sequences form the forecast. The forecast computed for this example for the set of possible values (e,s) is $(0.85, 0.15)$. Next we predict the most likely consensus answer based on second, third and fourth workers' reports. In this example, the most likely answer is e, since the other workers reported the sequence $\{s, e, e\}$. The first worker is rewarded $ln(0.85)$ based on the likelihood of answer e in the forecast when the logarithmic rule is used to calculate payments.

We use the simplified Galaxy Zoo example to demonstrate the fairness properties of consensus prediction payments. When normalized payments are computed with this rule, the payment vector is $(1, 0, 1, 1)$. As shown by this example, the reward of workers are not affected by the erroneous reports as long as the system can predict the correct answer accurately based on other workers' reports.

We present a formal definition of the consensus prediction rule. Let t be a consensus task, r be the sequence of worker reports collected for the task, and r_{-i} be the sequence excluding worker i's report. $\hat{A}_{-i}$ is a random variable for the consensus answer decided by the system if the system runs without access to worker i. In defining consensus prediction payments, we assume that a worker's inference is stochastically relevant for $\hat{A}_{-i}$ given feature set f. This is a realistic assumption because an inference of a worker provides evidence about the task, its correct answer, and other workers' inferences, which are used to predict a value for $\hat{A}_{-i}$.

Proposition 2. *For a given consensus task t and policy π, let $\hat{a}_{-i}$ be the consensus answer predicted based on r_{-i}. $M = (t, \pi, \tau^c)$ is strict Bayesian-Nash incentive compatible for any worker i, where*

$$\tau_i^c(r_i, r_{-i}) = S(p^c, \hat{a}_{-i}), \text{ where}$$
$$\text{for all } a_k \in A,\ p_k^c = Pr_f(\hat{A}_{-i} = a_k | C_i = r_i)$$

Proof. Under the assumption that C_i is stochastically relevant for $\hat{A}_{-i}$ given f, the proof follows from the definition of proper scoring rules. □

4.3.1 Calculating Consensus Prediction Payments

Next, we demonstrate the way payments can be calculated with the consensus prediction rule for consensus tasks in the equilibrium when all workers report their true inferences. The calculation of τ_i^c payments is a two step process; generating a forecast about $\hat{A}_{-i}$ based on worker i's report, and calculating a value for $\hat{a}_{-i}$ based on r_{-i}.

To generate a forecast for $\hat{A}_{-i}$, we simulate the consensus system for all possible sequences of worker reports that reach a consensus about the correct answer. $L_\emptyset$ is defined as the set of all such sequences. For any sequence r' in $L_\emptyset$, $M_\pi(r', f)$ is the consensus answer decided based on reports in r'. For each r', we calculate $Pr_f(r'|r_i)$, the likelihood of report sequence r' conditional on the fact that worker i already provided report r_i for the same task. $Pr_f(\hat{A}_{-i} = a|C_i = r_i)$ is computed as the cumulative probabilities of all $r' \in L_\emptyset$ that converge to answer a. For any value of $a \in A$ and $r_i \in \Omega_R$, $Pr_f(\hat{A}_{-i} = a|C_i = r_i)$ is computed as given below:

$$Pr_f(\hat{A}_{-i} = a|C_i = r_i) = \sum_{r' \in L_\emptyset} Pr_f(r'|r_i)\, \mathbf{1}_{\{a\}}(M_\pi(r', f))$$

We use the report of worker i as a feature to predict the likelihood of a report sequence $r' \in L_\emptyset$. Using the Bayes rule, $Pr_f(r'|r_i)$ is calculated as given below:

$$Pr_f(r'|r_i) \propto \sum_{a^* \in A} M_A(a^*, f_t)\, M_R(r_i, a^*, f_i) \prod_{l=1}^{|r'|} M_R(r_l, a^*, f_l)$$

The second step of τ_i^c calculation is predicting the realized value for $\hat{A}_{-i}$ based on r_{-i}, the actual set of reports collected from workers excluding worker i. $\hat{a}_{-i}$, the most likely value for $\hat{A}_{-i}$ based on r_{-i}, is calculated as follows: If there exists a substring of r_{-i} that starts with the first element of r_{-i} and converges on an answer, $\hat{a}_{-i}$ is assigned the value of this answer. Otherwise, calculating $\hat{a}_{-i}$ requires simulating all report sequences that start with r_{-i} and reach a consensus on the correct answer. $L_{r_{-i}}$ is the set of such sequences. $\hat{a}_{-i}$ is the answer that is most likely to be reached by the report sequences in $L_{r_{-i}}$.

$$\hat{a}_{-i} = \operatorname*{argmax}_{a \in A} \sum_{r' \in L_{r_{-i}}} Pr_f(r'|r_{-i})\, \mathbf{1}_{\{a\}}(M_\pi(r', f))$$

4.3.2 Calculating Consensus Prediction Payments Efficiently

Calculating payments with the consensus prediction rule is computationally more expensive than computing other payment rules introduced in this paper, as an iteration over exponential number of report sequences is required. The bottleneck of this computation is the calculation of $Pr_f(\hat{A}_{-i}|C_i = r_i)$. We approximate this value efficiently by using importance sampling. Let X be a random variable for the value of $Pr_f(\hat{A}_{-i} = a|C_i = r_i)$. Sampling a

report sequence $r' \in L_\emptyset$, such that the likelihood of the sample is proportional to $h(r') = Pr_f(r'|r_i)$, takes linear time in the length of r'. After sampling n report sequences $r'_1, ..., r'_n$, the expected value of X is computed as $\mu = \Sigma_{t=1}^{n} g(r'_t)$, where $g(r'_t) = \mathbf{1}_{\{a\}}(M_\pi(r'_t, f))$, and the variance is computed as $\sigma^2 = Var_h(g(r'))/n$. Let ϵ_s be a constant. We define λ_s as the likelihood that the error in calculating $Pr_f(\hat{A}_{-i} = a|C_i = r_i)$ exceeding constant ϵ_s. Using Chebyshev's inequality, we can calculate n, the number of samples needed to bound λ_s, as $n \leq \sigma^2/\lambda_s$.

In Section 6, we empirically evaluate the effect of calculating consensus prediction payments with this approximation on the truth promoting behavior of this rule.

4.4 Analysis of Payment Rules

The consensus prediction payment rule incentivizes workers to report truthfully under two conditions; (1) worker and answer models are common knowledge among the system and the workers, (2) a worker's inference (C_i) is stochastically relevant to $\hat{A}_{-i}$, the consensus answer that would be decided by the system without this worker's inference. We revisit the Galaxy Zoo example to demonstrate the incentive compatibility properties of consensus prediction payments. In the Galaxy Zoo example given earlier, all workers are equally competent in predicting the correct answer of a task. A worker inferring the correct answer of a galaxy as s increases the likelihood that the correct answer being s and also the likelihood of other workers inferring s. Consequently the worker's inference changes the likelihood of the value of $\hat{A}_{-i}$, which satisfies the stochastic relevance requirement. Given the common knowledge assumptions, the system can best predict $\hat{A}_{-i}$ if the worker reports truthfully. Thus, a worker maximizes her payment by reporting truthfully, even when she infers the unlikely answer, when other workers are reporting truthfully. The same reasoning can be used for worker populations including workers of varying competencies. For example, a system may have access to a low ratio of expert workers that can predict the correct answer with high accuracy and a larger ratio of workers that can barely do better than random. When the common knowledge assumption is satisfied, the system is able to distinguish competent workers from incompetent workers and calculate payments accordingly. For example, the influence of an expert's inference on predicting the system's likelihood of the correct answer and on predicting other workers' inferences would be different than the influence of a non-expert's inference. In such a domain, as long as the common knowledge assumptions are satisfied and the system can distinguish expert and non-expert workers, all workers are incentivized to report truthfully regardless of their relative ratios.

A consensus system may implement different policies from simple to complicated to decide on a consensus answer. The policy implemented in the system is used in the calculation of consensus prediction payments. This may raise a question about whether the implemented policy may affect the behavior of workers. The policy is used to calculate the signal for evaluating worker i's

report (i.e., the realized value of $\hat{A}_{-i}$, the answer that would be decided by the system without worker i's report). We will show that a worker cannot affect the evaluation signal $\hat{A}_{-i}$ with its report to the system, regardless of the policy implemented. Given that worker and answer models are common knowledge, a worker may affect $\hat{A}_{-i}$ only by influencing r_{-i}, the sequence of worker reports obtained from workers other than i. We will consider the approaches a worker may take to influence r_{-i}; (1) by influencing the workers that are hired by the system, and (2) by influencing the number of workers hired by the system. Given the definition of the policy, the system does not control who is hired next, so a worker cannot influence the workers that are hired. Moreover, the prediction of $\hat{A}_{-i}$ is independent of the number of workers hired by the system, as this calculation considers report sequences of any lengths that converge on an answer. Thus, a worker cannot influence the evaluation signal, regardless of the policy implemented. Due to the proper scoring rules used in payment calculations, a worker's expected payment depends on how well the realized value of $\hat{A}_{-i}$ can be predicted based on the worker's report. Under the assumption that worker and answer models are common knowledge and other workers are reporting truthfully, the worker maximizes her expected payment always by reporting truthfully, regardless the policy implemented. The same reasoning can be used to conclude that the implemented policy does not affect the behavior of workers when peer prediction rules are used to incentivize workers.

The consensus prediction payment rule may have practical advantages over the peer prediction rule due to its better fairness properties. Imagine a difficult task for which only a few number of competent workers can predict the correct answer. A system requires competent workers for solving such a task. When the peer prediction payment rule is implemented, a competent worker may receive a payment that is only as much as the payment of an incompetent worker, which may discourage the competent worker from participating. When the system implements consensus prediction payment, the payment of a competent worker is likely to be higher than the payment of an incompetent worker, if the system can deduce the correct answer and has accurate worker models. Thus, the system implementing consensus prediction payments is more likely to attract high quality workers and discourage low quality workers, which results in higher efficiencies for the system and the task owner.

An advantage of the peer prediction and consensus prediction payment rules is that they can adapt to changing worker populations with updating worker models in real-time as they make new observations about workers. For example, a group of malicious workers may collude on a strategy to increase their payments in a consensus system. Although these workers may initially succeed, the system can update the worker models as it makes observations about these workers. When the worker models can model the behavior of these workers properly, these workers may start getting penalized for not reporting honestly to the system.

This paper focuses on the challenge of incentivizing workers to report truthfully to a consensus system once they decide to participate in the system. A consensus system may face additional challenges in real-world applications in

terms of attracting workers. For example, the expected payment of a competent worker may be lower for a difficult task. The system may not be able to solve the task due to not being able to attract competent workers. Another challenge may arise if workers' expected payments vary depending on when they participate in the system. A worker may decide to wait to participate in the system which may reduce the efficiency of the system. An advantage of the payment rules that employ proper scoring rules is that the expected payment of a worker can be scaled to any desired value without degrading the incentive compatibility properties of these rules. It is a challenge for future work to develop methods for appropriately scaling payments in real-world applications to overcome difficulties in attracting workers.

5 Real-World Considerations

Systems implementing peer prediction and consensus prediction rules are incentive compatible under strong common knowledge assumptions. For a consensus system to have incentive compatibility, the prior probabilities on answers and the likelihoods of worker reports (conditioned on the task and the set of workers) should be common knowledge. This assumption is the biggest limitation in using these payment rules in real-world systems. We now explore approaches for relaxing these assumptions.

It is not realistic in many real-world settings to expect that workers of a system will have enough information about tasks and workers to accurately estimate prior probabilities on answers and the likelihood of worker reports. This situation clearly violates the common knowledge assumptions. One simple way to relax these assumptions is building trust between the system and the workers (e.g., via transparency of predictive models). As long as workers trust the system to calculate peer prediction or consensus prediction payments correctly, it is the best response for workers to reveal their true inference about a correct answer.

It is generally assumed that a system has enough history to learn prior answer probabilities and worker report probabilities. This history needs to be collected from truthful workers so that the system can learn about the true inferences of workers, and these models can be used for payment calculations. This requirement raises a question: How history data is collected from truthful workers without an incentive-compatible system in place? To address this question, there has been recent work on collecting truthful reports when participants of a system have private beliefs [22]. This work proposes a two-step revelation approach in which a participant reveals her belief before and after receiving a signal (experiencing a product or answering to a consensus task). The system uses the difference in these beliefs to infer the true report of the worker. The two-step revelation approach can be used with both the peer prediction and consensus prediction rules to promote truthful reporting when common knowledge assumptions do not hold. Having two-step revelation over beliefs clearly increases the reporting cost of a participant, but offers a viable approach to

collect enough data about workers' inferences until the system is able to train accurate predictive models.

Next, we address a more complicated case in which the system does not know about some features of the task at hand or the workers hired for the task, and thus cannot calculate payments accurately. As stated in previous sections, the incentive compatibility of consensus systems depends on whether payments can be computed accurately. Because payments are computed based on the predictions of predictive models, doing so not only requires having accurate models, but also having comprehensive set of evidences and features that can perfectly model a task and workers reporting for the task. If a system does not know some of the features that workers know, the common knowledge assumptions may not hold. For example, if a system cannot judge how difficult a task is, but a worker can, the worker may strategize to improve her payment by not reporting truthfully. The proposition below shows that when workers and the system have a channel to communicate, peer prediction and consensus prediction rules incentivize workers to communicate the difficulty of the task (or any other feature in f that the worker knows but the system does not) so that the common knowledge assumptions are satisfied and the system can accurately calculate payments.

We define two sets of features $\mathcal{F}_i^w$, $\mathcal{F}_i^s$ such that $\mathcal{F}_i = \mathcal{F}_i^w \cup \mathcal{F}_i^s$. $\mathcal{F}_i^s$ is the set of features that the system can infer correctly. This set may include the general statistics about the worker population and the tasks. $\mathcal{F}_i^w$ is the set of features that workers can infer correctly, but the system may not. This set may include the personal competency of worker i, whether the given task is relevant to the worker, and how difficult the task is for the worker. We define f_i^s as the true valuation of $\mathcal{F}_i^s$, f_i^w as the true valuation of $\mathcal{F}_i^w$, and $\bar{f}_i^w$ as the system's estimation of the features in $\mathcal{F}_i^w$. We assume that $\mathcal{F}_i^w$ is stochastically relevant for C_j for any worker j conditional on f_i^s and any realization of C_i (i.e., knowing the true value for these features help to better predict other workers' reports). We show below that if a system is implementing peer prediction rules, it is the equilibrium of the system for every worker i to report f_i^w as well as her true inference about the correct answer.

Proposition 3. *It is a strict Bayesian Nash equilibrium of $M = (t, \pi, \tau^p)$ for each worker i to report her inferences about $\mathcal{F}_i^w$ truthfully in addition to her report about the correct answer.*

Proof. We will use the logarithmic scoring rule in this proof for ease of representation. We show that the expected payment of worker i increases if she chooses to report f_i^w rather than not reporting. Let $V_i(r_i, f_i^w)$ be the expected payment of worker i when she reports r_i and f_i^w. We use $Pr(c_j|c_i, f_i^w \cup f_i^s)$ as a shorthand for $Pr(C_j = c_j|C_i = c_i, \mathcal{F} = f_i^w \cup f_i^s)$.

$$V_i(r_i, f_i^w) - V_i(r_i, \emptyset) = \sum_{c_j} (Pr(c_j|c_i, f_i^w \cup f_i^s)\ ln(Pr(c_j|c_i, f_i^w \cup f_i^s))) -$$
$$\sum_{c_j} Pr(c_j|c_i, f_i^w \cup f_i^s)\ ln(Pr(c_j|c_i, \bar{f}_i^w \cup f_i^s)$$
$$= D_{KL}(Pr(c_j|c_i, f_i^w \cup f_i^s)||Pr(c_j|c_i, \bar{f}_i^w \cup f_i^s)))$$

The difference in expected payments is the weighted KL divergence between the probability distribution when worker i's inference about $\mathcal{F}_i^w$ is revealed and when it is not revealed. Given that $\mathcal{F}_i^w$ is stochastically relevant for C_j, the KL divergence between these two distributions is always positive. Thus, $V_i(r_i, f_i^w) > V_i(r_i, \emptyset)$. □

Proposition 3 holds for the consensus prediction rule under the assumption that $\mathcal{F}_i^w$ is stochastically relevant for $\hat{A}_{-i}$ conditional on f_i^s and any realization of C_i. This stochastic relevance assumption is realistic because knowing the true values of the features in $\mathcal{F}_i^w$ help to better predict the correct answer of a task and the way workers report for the task, and thus help to predict $\hat{A}_{-i}$.

6 Empirical Evaluation

This section presents empirical evaluation of different payment rules when they are implemented in a system for solving consensus tasks. Our experiments focus on three main points; (1) understanding the way strategic reporting may hurt performance if the system is not incentive-compatible (2) comparing the fairness properties of peer prediction and consensus prediction rules, (3) studying the way peer prediction and consensus prediction rules behave when common knowledge assumptions do not hold.

To evaluate different payment rules in varying conditions, we developed a simulation system. The system takes as input consensus tasks with n possible answers. Prior probabilities over answers are generated randomly for each task. The correct answer is selected randomly such that the likelihood of an answer being correct is proportional to the answer's prior probability. Workers are assigned competency values that represent the likelihood that they will provide the correct answer. Competency values are sampled from a Gaussian distribution with a mean representing the population competency and a variance that is large enough to allow incompetent workers in a highly competent population and vice versa. A worker with minimum competency of 0 randomly selects an answer among n possibilities, and a worker with maximum competency of 1 always infers the correct answer. For any worker with a competency value that lies between 0 and 1, the likelihood of inferring the correct answer is linearly interpolated in $[1/n, 1]$. Neither workers, nor the system know workers' individual competencies. The system terminates a task when it reaches 95% confidence on an answer. For the first set of experiments, the prior probabilities on answers

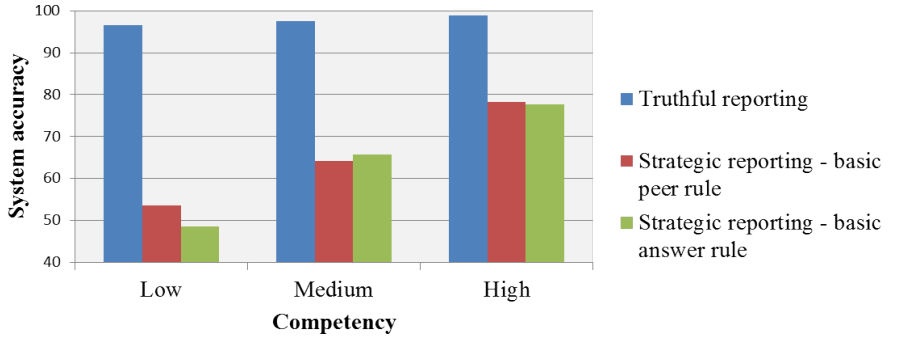

Figure 1: Effect of strategic reporting on system accuracy when basic payments are implemented.

and the average competency of the population are common knowledge between the system and the workers. Later in this section, we experiment when this condition does not hold.

In all experiments, workers are strategic. A strategic worker reports truthfully if reporting truthfully is an equilibrium of the system. Otherwise, she calculates the set of pure strategies that are equilibriums of the system and randomly picks a strategy from this set to follow.

The first set of experiments study the effect of truthful reporting on the system's performance. In these experiments, we randomly generated 10000 tasks for each experimental condition, and we vary the population competency between low (0.2), medium (0.5) and high (0.8). Figure 1 compares the system's performance in predicting the correct answer when the peer prediction or the consensus prediction rule is implemented and all workers are incentivized to report truthfully, and when basic payments are implemented and workers strategize about what to report. These results show that when a system is not incentive compatible, its performance may be significantly degraded by strategic reporting. The effect is more significant for systems recruiting workers of low competency in that the accuracy of the system is nearly halved due to strategic reporting. The effect of strategic reporting may become even more severe if workers find a way to coordinate on an equilibrium when truthful reporting is not an equilibrium.

Given the analysis of the accuracy degrading effect of strategic reporting on consensus tasks, we provide an analysis of truth promoting payment rules through the rest of the section. In the first set of experiments, we assume that common knowledge assumptions hold and thus all workers follow the equilibrium strategy of truthful reporting. In this equilibrium of truthful reporting, we focus on understanding the fairness properties of peer prediction and consensus prediction rules. We define fairness as providing rewards to workers who provide correct reports. We define a fairness metric, *absolute fairness* as the correlation

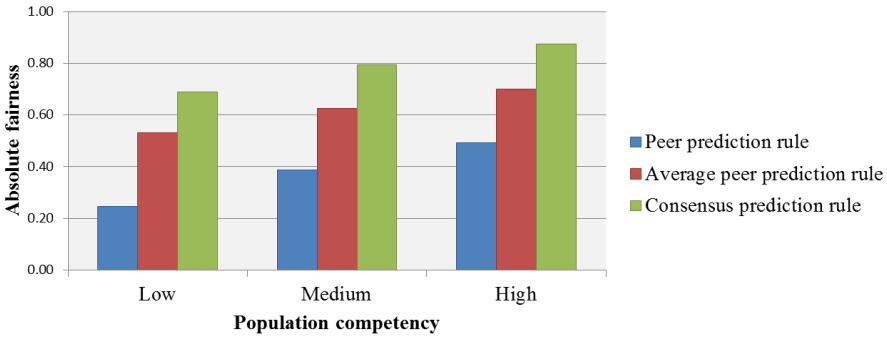

Figure 2: The comparison of fairness of truth promoting payment rules.

between correctness of a worker's report and the corresponding payment [3]. A system with high absolute fairness offers a fair value to all workers that provide the correct answer of a task and thus contribute to the system. In such a system, there would be less envy among people that all provide the correct answer of a task.

Figure 2 compares the absolute fairness of payment rules. The results show that the peer prediction rule has the worst fairness properties. When peer prediction payments are distributed, a worker providing the correct answer of a task may unfairly receive a lesser payment compared to other workers providing the correct answer for the same task or other tasks. The fairness properties of the average peer prediction rule is better than the peer prediction rule, but is worse than the consensus prediction rule. When the average peer prediction rule is implemented, a worker with the correct report receives the same amount as other workers reporting correctly for the same task, but she may receive a lesser amount compared to other workers reporting correctly for other tasks. The consensus prediction rule outperforms other rules in terms of fairness for all competency levels because it rewards workers consistently depending on how well their reports contribute to the answer that is predicted by the system.

Next, we study the way payments computed with the peer and consensus prediction rules incentivize workers when common knowledge assumptions do not hold and thus the payment calculations are noisy. We add noise sampled from a Gaussian distribution to each prediction obtained by the system from the answer and report models for calculating payments. We perform a worst-case analysis such that workers can observe the true probabilities and also the noisy probabilities used by the system, all workers are strategic, and they are computationally powerful to compute peer prediction and consensus prediction rules. In this worst case setting, we calculate the percentage of strategic workers that deviate from reporting truthfully. The population competency varies between low and high competency. The consensus prediction rule is computed

[3] All correlations are statistically significant with p=0.01.

with sampling ($\epsilon_s = 0.1, \delta_s = 0.01$), which introduces another type of noise to the consensus prediction payment calculations. We vary the magnitude of noise by varying the standard deviation of the Gaussian distribution σ between no noise ($\sigma = 0$), low noise ($\sigma = 0.01$), medium noise ($\sigma = 0.05$), high noise ($\sigma = 0.1$) and very high noise ($\sigma = 0.2$).

Figure 3 reports the ratio of truthful reporting for peer prediction and consensus prediction rules, and for basic payment rules when varying levels of noise are added to the predictive models used in payment calculations. These results show that the incentive compatibility of both peer prediction and consensus prediction rules are robust to low levels of noise in predictive models. In addition, the consensus prediction rule is robust to low levels of noise introduced by the approximate calculation of these payments. When the noise increases to high levels, the ratios of truthful reporting for these rules degrade slightly. However, despite the increasing noise, the results show that these rules are significantly better at promoting truthful reporting than basic payments even when the noise level is very high.

Figure 3 presents promising results about the performance of peer prediction and consensus prediction payment rules even when common knowledge assumptions do not hold. These experiments employ a simple policy in payment calculations that does not reason about the noise in predictive models while deciding on consensus. Because the consensus prediction payments use the consensus answer derived by the system as the signal for evaluating worker reports, the robustness of this payment rule can be further improved with better policies that can reason about the noise in predictive models in its decision-making process. When coupled with such policies, the robustness of the consensus prediction rules may outperform the robustness of the peer prediction rules. It is important to note that the truth promoting properties of these rules are likely to be better in real-world systems in which the worst case assumptions presented above do not necessarily hold. In settings in which workers are computationally bounded or they cannot infer the noise in the system, it may be impractical for workers to strategize over a system implementing peer prediction or consensus prediction payments. In addition, as demonstrated by Proposition 3, the system may be able to correct the noise in predictive models by establishing communication and collaboration with workers.

To the best of our knowledge, this evaluation is the first empirical study of the behavior of peer prediction rules and consensus prediction rules when common knowledge assumptions do not necessarily hold. We believe that understanding the behavior of different payments rules in real-world systems with noise and computational limitations will help to determine which payment rule is more suitable for a given domain, and thus will foster the application of these ideas in diverse settings.

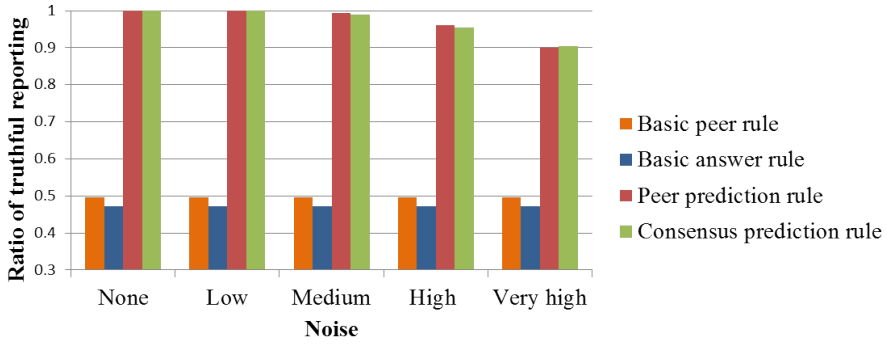

Figure 3: Worst-case analysis of the incentive compatibility of payments for varying levels of noise.

7 Future Work and Conclusions

We reviewed key opportunities and challenges for developing truthful, fair, and computationally feasible incentive mechanisms for crowdsourcing. We also studied the issues that arise in applying peer prediction and consensus prediction rules in real-world systems. We believe that the results in this paper pave the way for future work in the directions of formalized approaches for more efficient crowdsourcing systems, and application of peer prediction and consensus prediction rules to diverse real-world systems.

Future work on crowdsourcing includes designing truthful incentive mechanisms for a larger variety of tasks, including opinion tasks and tasks that may require complex workflows, and developing mechanisms to prevent collusion among workers [10]. Research directions on prediction rules include the pursuit of new approaches for relaxing common knowledge assumptions and studying the properties of these payment rules in other real-world settings. We believe that the use of truthful and fair mechanisms promises to enhance the operation of crowdsourcing for both authors and contributors, and can promote the wider use of such systems as a trusted methodology for problem solving.

References

[1] M. Bernstein, G. Little, R. Miller, et al. Soylent: a word processor with a crowd inside. In *symposium on User interface software and technology*, 2010.

[2] Y. Chen and D. Pennock. Designing markets for prediction. *AI Magazine*, 2010.

[3] P. Dai, Mausam, and D. Weld. Decision-theoretic control of crowd-sourced workflows. In *AAAI*, 2010.

[4] J. Douceur and T. Moscibroda. Lottery trees: motivational deployment of networked systems. *ACM SIGCOMM Computer Communication Review*, 37(4):121–132, 2007.

[5] A. Ghosh, S. Kale, and P. McAfee. Who moderates the moderators? crowdsourcing abuse detection in user-generated content. In *Proceedings of the 12th ACM conference on Electronic commerce*, pages 167–176. ACM, 2011.

[6] S. Goel, D. Reeves, and D. Pennock. Collective revelation: a mechanism for self-verified, weighted, and truthful predictions. In *EC*, 2009.

[7] R. Hanson. Logarithmic market scoring rules for modular combinatorial information aggregation. *The Journal of Prediction Markets*, 1(1):3–15, 2007.

[8] J. Horton and L. Chilton. The labor economics of paid crowdsourcing. In *EC*, 2010.

[9] E. Huang, H. Zhang, D. Parkes, K. Gajos, and Y. Chen. Toward automatic task design: a progress report. In *SIGKDD Workshop on Human Computation*, 2010.

[10] R. Jurca and B. Faltings. Collusion-resistant, incentive-compatible feedback payments. In *EC*, 2007.

[11] E. Kamar, S. Hacker, and E. Horvitz. Combining human and machine intelligence in large-scale crowdsourcing. In *Proceedings of the 11th international joint conference on Autonomous agents and multiagent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2012.

[12] A. Kittur, E. Chi, and B. Suh. Crowdsourcing user studies with Mechanical Turk. In *SIGCHI conference on Human factors in computing systems*, 2008.

[13] C. Lintott, K. Schawinski, A. Slosar, et al. Galaxy Zoo: morphologies derived from visual inspection of galaxies from the Sloan Digital Sky Survey. *Monthly Notices of the Royal Astronomical Society*, 2008.

[14] W. Mason and D. Watts. Financial incentives and the performance of crowds. *ACM SIGKDD Explorations Newsletter*, 2010.

[15] K. Mieszkowski. "i make $1.45 a week and i love it", 2006.

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

[17] R. Nelson and D. Bessler. Subjective probabilities and scoring rules: experimental evidence. *American Journal of Agricultural Economics*, 1989.

[18] D. Prelec. A Bayesian truth serum for subjective data. *Science*, 2004.

[19] D. Shahaf and E. Horvitz. Generalized task markets for human and machine computation. In *AAAI*, 2010.

[20] V. Sheng, F. Provost, and P. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 614–622. ACM, 2008.

[21] L. Von Ahn and L. Dabbish. Designing games with a purpose. *Communications of the ACM*, 2008.

[22] J. Witkowski and D. C. Parkes. Peer Prediction with Private Beliefs. In *Proceedings of the Workshop on Social Computing and User Generated Content*, 2011.

[23] H. Zhang, E. Horvitz, Y. Chen, and D. Parkes. Task routing for prediction tasks. In *Proceedings of the 11th international joint conference on Autonomous agents and multiagent systems*. ACM, 2012.