Predicting Own Action: Self-Fulfilling Prophecy Induced by Proper Scoring Rules

Masaaki Oka[1], Taiki Todo[1], Yuko Sakurai[2], and Makoto Yokoo[1]
1: Kyushu University, Fukuoka, 819-0395, Japan
2: Kyushu University and JST PRESTO, Fukuoka, 819-0395, Japan

Abstract

This paper studies a mechanism to incentivize agents who predict their own future actions and truthfully declare their predictions. In a crowdsouring setting (e.g., participatory sensing), obtaining an accurate prediction of the actions of workers/agents is valuable for a requester who is collecting real-world information from the crowd. If an agent predicts an external event that she cannot control herself (e.g., tomorrow's weather), any proper scoring rule can give an accurate incentive. In our problem setting, an agent needs to predict her own action (e.g., what time tomorrow she will take a photo of a specific place) that she can control to maximize her utility. Also, her (gross) utility can vary based on an eternal event.

We first prove that a mechanism can satisfy our goal if and only if it utilizes a strictly proper scoring rule, assuming that an agent can find an optimal declaration that maximizes her expected utility. This declaration is *self-fulfilling*; if she acts to maximize her utility, the probabilistic distribution of her action matches her declaration, assuming her prediction about the external event is correct. Furthermore, we develop a heuristic algorithm that efficiently finds a semi-optimal declaration, and show that this declaration is still self-fulfilling. We also examine our heuristic algorithm's performance and describe how an agent acts when she faces an unexpected scenario.

Introduction

This paper considers a mechanism that gives an incentive to a worker to predict her own future action and perform it as she predicted. Recently, eliciting/aggregating information about uncertain events from agents is becoming a common research topic due to the expansion of prediction markets and services based on the wisdom of crowds (Conitzer 2009; Chen and Pennock 2010; Chen et al. 2011; Law and Ahn 2011; Wolfers and Zitzewitz 2004). Studies have designed incentive mechanisms for quality control mechanisms for crowdsourced tasks (Bacon et al. 2012; Cavallo and Jain 2013; Lin, Mausam, and Weld 2012; Sakurai et al. 2013; Witkowski and Parkes 2012). These works utilize the results obtained in the literature of mechanism design, which is a subfield of game theory and microeconomics that studies how to design mechanisms for good outcomes even when agents act strategically.

Prediction mechanisms aggregate probabilistic information from agents to accurately predict uncertain events. Proper scoring rules have been developed to incentivize an agent to truthfully reveal her subjective probability/belief (Brier 1950; Gneiting and Raftery 2007; Matheson and Winkler 1976; Savage 1971). If an agent predicts an external event that she cannot control herself (e.g., tomorrow's weather), and its outcome does not affect her (except for the reward she obtains from a correct prediction), any proper scoring rule can give a correct incentive. Several researchers have investigated more general cases where the prediction and actions of an agent or a principal interact with each other (Boutilier 2012; Chen and Kash 2011; Othman and Sandholm 2010; Shi, Conitzer, and Guo 2009). We discuss more details about the differences between these works and our problem setting in the next section.

In a typical human computation/crowd sourcing setting, a requester tries to obtain contributions from an online community. For example, recruiting workers is an important issue in participatory sensing where a requester gathers real-world information from the crowd (Reddy, Estrin, and Srivastava 2010). If a worker declares the time she will work in advance and completes her task as declared, the requester benefits and can make an efficient recruiting plan. In this paper, we consider the following situation. A requester asks each worker to predict her action tomorrow (e.g., what time tomorrow she will take a photo of an object at a specific location). The worker declares her probabilistic prediction (e.g., a 90% chance that she will work in the daytime and a 10% chance at night). We assume her predictions are probabilistic, since her utility depends on an external event (e.g., tomorrow's weather). Based on the prediction, the provider plans a reward plan (e.g., a 20% bonus if she works in the daytime). The next day, she learns the outcome of the external event and decides her action based on the reward plan as well as the external event.

Crowdsoucing requesters who receive contributions from workers are not the only people who want to know the future behaviors of many people. Imagine that a service provider wants to obtain a prediction about the future actions of his customers. In this case, he can be considered a requester in a human computation process that elicits information from the crowd. One major motivation for considering this setting is that for a certain provider (e.g., electricity or shuttle bus services for big events), providing adequate supply to meet customer demand is crucial. If the provider prepares sufficient resources in reserve, such a disaster can be avoided but at a cost. The provider gladly offers incentives to customers if he can accurately predict and save the expense of unnecessary reserves. If customers aren't offered incentives, they are probably unwilling to give accurate predictions to the provider.

The above rather complicated problem settings introduce several new research issues. First, an agent needs to compute her optimal declaration and strategy to maximize her utility by considering possible reward plans. This is a complicated optimization problem. For requesters, it is not obvious what kind of reward plans are effective to elicit truthful declarations.

We first prove that if a mechanism utilizes a strictly proper scoring rule, then it can elicit a *self-fulfilling* declaration, assuming an agent can find an optimal declaration that maximizes her expected utility and that her prediction of the external event is correct. The declaration is self-fulfilling if she acts to maximize her utility and if the probabilistic distribution of her actions matches her declaration. Furthermore, we show that if a mechanism does not utilize a strictly proper scoring rule, then there exists a case where the declaration that maximizes an agent's expected utility is not self-fulfilling.

Next, we examine how an agent can solve her optimization problem, which is very difficult, since there exist an infinite number of possible strategies and declarations. We first show that for each agent, declaring truthfully is always optimal. Furthermore, we show that an optimal strategy is always in a subclass of strategies and finite. Thus, she can obtain an optimal strategy by enumerating this subclass. However, its size can be exponentially large to reflect the number of alternative actions and possible outcomes of external events. We develop an iterative improvement type algorithm for efficiently finding a semi-optimal declaration and a strategy and show that this semi-optimal declaration remains self-fulfilling.

We also examine the performance of our heuristic algorithm, and describe by computational simulations how an agent acts when she faces an unexpected scenario. Our results illustrate that the expected utility obtained by the heuristic algorithm is more than 92% of the optimal case where the expected utility is maximized. The agent still behaves as expected in an unexpected scenario when the reward provided by a rule is sufficiently large.

The rest of this paper is organized as follows. First, we explain related works and introduce a model of our problem setting. Next, we characterize our mechanism in which the requesters/principals ask each agent to predict her expected probability distribution over possible actions. Then, we present our heuristic algorithm for a semi-optimal declaration and provide simulation results for evaluating it.

Related works

There exists a vast amount of work related to proper scoring rules in AI literature. Here, we explain several existing works that are related to our problem setting, i.e., those considering the interaction between predicted events and actions. Shi, Conitzer, and Guo (2009) considered a situation where experts, once they report their forecasts, can take actions to alter the probabilities of the outcomes in question. In particular, they can take undesirable actions for the principal, e.g., a worker intentionally delays a project's completion time to fulfill her prediction. Unlike our model, the expert's utility is determined exclusively by the payoff offered by the principal.

Next, Othman and Sandholm (2010) provided the first explicit, formal treatment of a principal who makes decisions based on expert forecasts. They addressed several key difficulties that arise due to the conditional nature of forecasts. Chen and Kash (2011) extended this model to a wider class of informational settings and decision policies.

Boutilier (2012) considered a case where the principal decides based on the prediction of experts, where the decision affects the utility of the experts (e.g., a stockpiling medicine policy as a safeguard against influenza benefits pharmaceutical companies). Our model is different from these since we assume that the principal wants to know the action of each agent, whose utility is not affected by the actions of the principal.

Model

In this section, we explain the model of our problem setting.

Definition 1 (Actions). *An agent has set of actions $M = \{1, \ldots, m\}$ and must choose exactly one action $i \in M$ the next day.*

M is a set of alternatives that the agent can choose. If she is deciding what time tomorrow she will take a photo of an object at a specific location, each i represents the possible time slots she can choose: either "daytime" or "night."

Definition 2 (Scenarios). *There is set of scenarios $N = \{1, \ldots, n\}$. The nature chooses exactly one scenario $j \in N$ the next day.*

A set of scenarios represents an external event, and each scenario $j \in N$ represents its outcome. For example, if the external event is tomorrow's weather, possible scenarios include "sunny," "cloudy," or "rainy."

We assume an agent knows the probability distribution for the set of scenarios.

Definition 3 (Scenario probability distribution). *A scenario probability distribution is a n-element column vector $p = \begin{pmatrix} p_1 \\ \vdots \\ p_n \end{pmatrix}$, where p_j is the probability that scenario j happens. $\forall j, 0 \leq p_j \leq 1$ and $\sum_{j \in N} p_j = 1$ hold.*

Thus, $p = \begin{pmatrix} 0.7 \\ 0.2 \\ 0.1 \end{pmatrix}$ means that the agent considers the "sunny" probability to be 70%, the "cloudy" probability to be 20%, and the "rainy" probability to be 10%.

We assume that the gross utility of an action is affected by the choice of scenarios. The collection of these utility values is represented as a matrix.

Definition 4 (Gross utility matrix). *The gross utility of an agent is defined as $m \times n$ gross utility matrix:*

$$G = \begin{pmatrix} g_{1,1} & \cdots & g_{1,n} \\ \vdots & \ddots & \vdots \\ g_{m,1} & \cdots & g_{m,n} \end{pmatrix},$$

where $g_{i,j}$ is a real value that represents the gross utility of taking action i when the scenario is j.

We also represent G as an array of column vectors, i.e., $G = (g_1, \ldots, g_n)$, where each g_j represents the gross utility vector when scenario j occurs.

Let us show an example of G in our participatory sensing example:

	sunny	cloudy	rainy
daytime	10	5	0
night	0	5	8

Here, the agent prefers taking photos in the daytime if the weather is sunny, but she prefers working at night if it is rainy. If the weather is cloudy, she has no preference between daytime or night.

We assume that an agent is risk neutral and that her utility is quasi-linear, i.e., if her utility is the sum of (i) the gross utility of an action and (ii) her reward from the requester.

Consider a m-element column vector

$$\pi = \begin{pmatrix} \pi_1 \\ \vdots \\ \pi_m \end{pmatrix},$$

where $\forall i, 0 \leq \pi_i \leq 1$ and $\sum_{i \in M} \pi_i = 1$ hold. Thus, π represents a probability distribution over M. Let Π denote the set of all possible probability distributions over M.

Next, we define an agent's strategy. For each scenario, the agent needs to decide which action to choose. We assume the choice of an action can be non-deterministic, i.e., flipping a coin. A strategy for each scenario j is represented as $s_j \in \Pi$.

The entire strategy for all scenarios is defined as follows.

Definition 5 (Strategy matrix). *An agent's strategy is described as an array of column vectors*

$$S = (s_1, \ldots, s_n),$$

where each $s_j \in \Pi$. As a whole, a strategy is represented as an $m \times n$ matrix:

$$S = \begin{pmatrix} s_{1,1} & \cdots & s_{1,n} \\ \vdots & \ddots & \vdots \\ s_{m,1} & \cdots & s_{m,n} \end{pmatrix}.$$

Let $\mathbf{S}$ denote the set of all possible strategies. If the agent commits to S, her expected probability distribution over the possible actions is given as Sp.

Let us show an example of S in our participatory sensing example.

	sunny	cloudy	rainy
daytime	1.0	0.5	0
night	0	0.5	1.0

Here, the agent is planning to take photos in the daytime if the weather is sunny, but at night if it is rainy. If the weather is cloudy, she will flip a coin. Based on this S,

$$Sp = S \begin{pmatrix} 0.7 \\ 0.2 \\ 0.1 \end{pmatrix} = \begin{pmatrix} 0.8 \\ 0.2 \end{pmatrix}.$$

An agent can calculate an expected gross utility for strategy S.

Definition 6 (Expected gross utility). *We define $G \otimes S$ as an n-element row vector,*

$$G \otimes S = (v_1, \ldots, v_n)$$

where $v_j = g_j \cdot s_j$. Here, each v_j represents the gross utility for strategy S when scenario j occurs. Therefore, $(G \otimes S)p$ gives the expected gross utility for strategy S.

Thus, we obtain: $G \otimes S = (10, 5, 8)$,

and $G \otimes S \begin{pmatrix} 0.7 \\ 0.2 \\ 0.1 \end{pmatrix} = 8.8.$

In our problem setting, the principal asks each agent to predict her expected probability distribution over the possible actions. She selects $\pi \in \Pi$ and declares π to the principal, who rewards to her based on her declaration π and her action choice determined by a reward function.

Definition 7 (Reward function). *Reward function $r(\cdot)$ takes a declaration as input and returns a reward vector:* $r(\pi) = \begin{pmatrix} r_1 \\ \vdots \\ r_m \end{pmatrix}$, *where each r_i is a real value that represents the reward for taking action i when declaring π.*

We assume reward function $r(\cdot)$ is fixed, which an agent knows when she determines her declaration.

Next, we explain strictly proper scoring rules and apply them as a reward function to satisfy the following condition.

Definition 8 (Strictly proper scoring rule). *$r(\cdot)$ is a strictly proper scoring rule, if $\forall \pi, \pi'$, if $\pi \neq \pi'$, the following condition holds:*

$$\pi \cdot r(\pi) > \pi \cdot r(\pi').$$

Here, $\pi \cdot r(\pi')$ represents the expected reward of an agent, when she declares π' and the principal presents reward vector $r(\pi')$, but she expects her actual action probability to be π. The above definition of a strictly proper scoring rule means that an agent can maximize her expected reward by truthfully declaring her expectation.

There exists a variety of strictly proper scoring rules. Here, we describe two representative examples. First, a spherical proper scoring rule is defined by

$$r_i = \alpha \frac{\pi_i}{\sqrt{\sum_{1 \leq k \leq m} \pi_k^2}}.$$

Second, a quadratic proper scoring rule is defined by

$$r_i = \alpha(2\pi_i - \sum_{1\le k\le m} \pi_k^2 + 1).$$

Here, α indicates the maximum amount of the scores.

Assume the agent declares $\pi = \begin{pmatrix} 0.8 \\ 0.2 \end{pmatrix}$, and the principal applies a spherical proper scoring rule with $\alpha = 5$, $r(\pi) = \begin{pmatrix} 4.85 \\ 1.21 \end{pmatrix}$.

Next, we calculate the expected (net) utility of a strategy and a declaration.

Definition 9 (Expected Utility). *Let $u(S,\pi)$ denote the agent's expected utility, when she declares π and commits to strategy S, which is calculated as follows:*

$$\begin{aligned} u(S,\pi) &= \sum_{j\in N} p_j(g_j \cdot s_j + r(\pi)\cdot s_j) \\ &= \sum_{j\in N} p_j(g_j\cdot s_j) + \sum_{j\in N} p_j(r(\pi)\cdot s_j) \\ &= (g_1\cdot s_1,\ldots,g_n\cdot s_n)p + (\sum_{j\in N} p_j s_j)\cdot r(\pi) \\ &= (G\otimes S)p + (Sp)\cdot r(\pi). \end{aligned}$$

Thus, the expected (net) utility can be divided into two parts, the expected gross utility and the expected reward. In our participatory sensing example,

$$u(S,\pi) = 8.8 + \begin{pmatrix} 0.8 \\ 0.2 \end{pmatrix} \cdot \begin{pmatrix} 4.85 \\ 1.21 \end{pmatrix} = 12.92.$$

Here, we introduce the properties related to declarations and strategies.

Definition 10 (Truthfulness). *S and π are truthful if $Sp = \pi$ holds.*

Definition 11 (Self-fulfillingness). *π is self-fulfilling if there exists strategy S, such that S and π are truthful and $\forall j \in N, \forall s'_j \in \Pi$,*

$$g_j\cdot s_j + r(\pi)\cdot s_j \ge g_j\cdot s'_j + r(\pi)\cdot s'_j$$

holds.

If π is self-fulfilling, the agent has no incentive to move from S even after she learns the outcome of tomorrow's actual scenario. Thus, on average, she is going to act as she predicted; her actual action probability distribution becomes Sp, which matches her declaration $\pi = Sp$.

In our participatory sensing example, it is clear that S and $\pi = \begin{pmatrix} 0.8 \\ 0.2 \end{pmatrix}$ are truthful, but π is not self-fulfilling. For example, if the weather is cloudy, the net utility becomes

$$\begin{pmatrix} 5 \\ 5 \end{pmatrix} + \begin{pmatrix} 4.85 \\ 1.21 \end{pmatrix} = \begin{pmatrix} 9.85 \\ 6.21 \end{pmatrix}.$$

Thus, the agent prefers to take photos in the daytime rather than flipping a coin.

Let us formally state the goals of the agent and the principal in our model.

Agent:

Today: Based on $G, p, r(\cdot)$, an agent finds

$$\pi^* = \arg\max_{\pi\in\Pi}(\max_{S\in\mathbf{S}} u(S,\pi)),$$

and declares it to the principal.

Tomorrow: Based on $r(\pi^*)$ and g_j, where j is the scenario chosen by the nature, an agent finds

$$s_j^* = \arg\max_{s_j\in\Pi}(g_j + r(\pi^*))\cdot s_j,$$

and acts based on s_j^*.

Principal:

- The principal must determine appropriate reward function $r(\cdot)$, such that if the agent can find optimal declaration π^*, then π^* is self-fulfilling.
- Even if the agent is unable to find an optimal declaration since her computational capability is limited, there exists an efficient way for the agent to find semi-optimal π', where π' is self-fulfilling.

Characteristics of Reward Mechanism

Next, we investigate the characteristics of our mechanism, in which the principal asks each agent to predict her expected probability distribution over tomorrow's possible actions. The principal rewards her based on her declaration and her actual action, assuming the principal uses a strictly proper scoring rule. Furthermore, we examine the problem each agent must solve to obtain an optimal strategy and declaration.

As described in the previous section, an agent first must find optimal declaration π^*, such that

$$\pi^* = \arg\max_{\pi\in\Pi}(\max_{S\in\mathbf{S}} u(S,\pi)).$$

For each agent, finding π^* is not easy. If we fix π, the net utility for each action is determined. Thus, for each possible scenario j, finding optimal strategy $s_j \in \Pi$ is easy; finding $\max_{S\in\mathbf{S}} u(S,\pi)$ is also easy. However, since there exists an infinite amount of π, we cannot exhaustively check all of them.

Assume an agent can somehow obtain π^*. Also, she chooses

$$S^* = \max_{S\in\mathbf{S}} u(S,\pi^*).$$

Now, we can prove that this declaration is self-fulfilling with strategy S^*.

Theorem 1. *If reward function $r(\cdot)$ is a strictly proper scoring rule, then $\pi^* = \arg\max_{\pi\in\Pi}(\max_{S\in\mathbf{S}} u(S,\pi))$ is self-fulfilling for any agent.*

Proof. First, we show that S^* and π^* are truthful. By contradiction, assume that $S^*p \ne \pi^*$ holds. Then, consider the utility of $u(S^*, S^*p)$; i.e., the agent's utility when using strategy S^* but declaring S^*p instead of π^*. Then, we obtain the following inequality by applying the property of strictly proper scoring rules.

$$\begin{aligned} & u(S^*, S^*p) - u(S^*,\pi^*) \\ &= (S^*p)\cdot r(S^*p) - (S^*p)\cdot r(\pi^*) > 0 \end{aligned}$$

As a result, we obtain $u(S^*, S^*p) > u(S^*, \pi^*)$. This contradicts how we chose π^* and S^*.

Next, by contradiction, assume $j \in N$ and $s_j \in \Pi$, such that $g_j \cdot s_j + r(\pi^*) \cdot s_j > g_j \cdot s_j^* + r(\pi^*) \cdot s_j^*$ holds. Then consider another strategy S which is basically the same as S^*, but its j-th column is replaced by s_j. Then, $u(S, \pi^*) > u(S^*, \pi^*)$ holds. This contradicts how we chose S^*. □

We can also prove that if π^* is a self-fulfilling declaration, then a reward function is a strictly proper scoring rule.

Theorem 2. *If reward function $r(\cdot)$ is not a strictly proper scoring rule, then there exists an agent such that $\pi^* = \arg\max_{\pi \in \Pi}(\max_{S \in \mathbf{S}} u(S, \pi))$ is not self-fulfilling.*

Proof. First, we assume rewards are non-negative and their maximum amount is bounded by α. Since r is not a strictly proper scoring rule, $\exists \pi' \neq \pi''$ such that

$$\pi' \cdot r(\pi') \leq \pi' \cdot r(\pi'')$$

holds. WLOG, we can assume

$$\pi'' = \arg\max_{\pi \in \Pi} \pi' \cdot r(\pi).$$

Now, consider a case with m scenarios whose probabilities are given as $p = \pi'$. The gross utility is given as G, where G is an $m \times m$ matrix where each diagonal element is $C >> \alpha$ and the other elements are 0. It is clear that for any π, best strategy S^* does not change, where S^* is an $m \times m$ identity matrix. It is clear that

$$\pi'' = \arg\max_{\pi \in \Pi} u(S^*, \pi) = \pi^*.$$

However, since $S^*p = \pi' \neq \pi''$, π'' is not truthful. As a result, π'' is not self-fulfilling. □

These theorems show that a reward function can elicit a self-fulfilling declaration if and only if it is a strictly proper scoring rule.

Note that an agent maximizes her expected net utility (i.e., her expected gross utility + rewards), but not necessarily her gross utility. Thus, our mechanism incurs some efficiency loss. However, as shown in Theorem 2, since a strictly proper scoring rule must be used to obtain a self-fulfilling declaration, such efficiency loss is inevitable. We guarantee that it is bounded by maximum reward α. At the end of this section, we show an example where an agent chooses an action that does not maximize her gross utility, since the provider gives a reward to incentivize her for making a self-fulfilling declaration.

We can generalize Theorem 1 as follows.

Theorem 3. *$\forall S \in \mathbf{S}, \forall \pi$, if $Sp \neq \pi$, the following condition holds:*

$$u(S, Sp) > u(S, \pi).$$

Proof. We obtain

$$u(S, Sp) - u(S, \pi) = (Sp) \cdot r(Sp) - (Sp) \cdot r(\pi).$$

Since the property of strictly proper scoring rules guarantees that this difference exceeds 0, we can obtain $u(S, Sp) > u(S, \pi)$. □

Theorem 3 means that if an agent commits to strategy S, she can automatically find the best declaration; truthful declaration is optimal. Instead of searching for $\pi^* = \arg\max_{\pi \in \Pi}(\max_{S \in \mathbf{S}} u(S, \pi))$, the agent can search for $S^* = \arg\max_{S \in \mathbf{S}} u(S, Sp)$. However, since $|\mathbf{S}|$ is infinite, it is impossible to enumerate all strategies.

To overcome this problem, we consider a subclass of $\mathbf{S}$, which consists of finite strategies.

Definition 12 (Deterministic strategy). *Strategy vector s_j is deterministic if it contains exactly one element whose value is 1, and the others are 0. S is also deterministic if all of its strategy vectors are deterministic.*

Let $\mathbf{S_d}$ be the set of all possible deterministic strategies.

The following theorem holds.

Theorem 4. *$\forall S \in \mathbf{S}, \forall \pi$, $\exists S' \in \mathbf{S_d}$, such that the following condition holds:*

$$u(S', \pi) \geq u(S, \pi).$$

Proof. When we fix π, the net utility (i.e., gross utility plus rewards) is determined. Then we can create S' as follows. For each $j \in N$, choose i so that i-th element of $g_j + r(\pi)$ is maximal. Set s_j as a deterministic strategy where the i-th element is 1 and the others are 0. Set S' to $(s_1, \ldots, s_n)$. Clearly, $u(S', \pi)$ is at least as good as $u(S, \pi)$ for any S. □

In our participatory sensing example, for $\pi = \begin{pmatrix} 0.8 \\ 0.2 \end{pmatrix}$, deterministic strategy S', which is defined as follows, outperforms original non-deterministic strategy S.

	sunny	cloudy	rainy
daytime	1.0	1.0	0
night	0	0	1.0

Here, $u(S', \pi) = 8.8 + \begin{pmatrix} 0.9 \\ 0.1 \end{pmatrix} \cdot \begin{pmatrix} 4.85 \\ 1.21 \end{pmatrix} = 13.29$, is larger than $u(S, \pi) = 12.92$.

This theorem means that, the agent does not need to use a non-deterministic strategy when her declaration is fixed. However, it is not obvious whether we can restrict our attention to deterministic strategies when we search for $S^* = \arg\max_{S \in \mathbf{S}} u(S, Sp)$, since the choice of S affects Sp, i.e., the optimal declaration for S. The choice of Sp affects the reward. Thus, perhaps by choosing a non-deterministic strategy, the agent can choose a better declaration and improve her expected utility.

The following theorem shows that this is not the case; an agent can restrict her attention to deterministic strategies to obtain S^*.

Theorem 5. *Let $\hat{S} = \arg\max_{S \in \mathbf{S_d}} u(S, Sp)$. Then, $\forall S \in \mathbf{S}, \forall \pi$, and the following condition holds:*

$$u(\hat{S}, \hat{S}p) \geq u(S, \pi).$$

Proof. By contradiction, assume that for S and π, $u(\hat{S}, \hat{S}p) < u(S, \pi)$ holds. From Theorem 4, there exists $S' \in \mathbf{S_d}$ such that $u(S', \pi) \geq u(S, \pi)$ holds. From Theorem 3, we obtain $u(S', S'p) \geq u(S', \pi)$. From these facts, we derive $u(S', S'p) > u(\hat{S}, \hat{S}p)$, contradicting the definition of $\hat{S}$. □

Heuristic algorithm

```
1.  Arbitrarily select initial $\pi$.
2.  For each $j \in N$, choose $i$ so that the $i$-th element
    of $g_j + r(\pi)$ is maximal.
    Set $s_j$ as a deterministic strategy where
    the $i$-th element is 1 and the others are 0.
    Set $S$ to $(s_1, \ldots, s_n)$.
3.  If $Sp = \pi$, return $S$.
    Otherwise, set $\pi \leftarrow Sp$ and go to 2.
```

Figure 1: Heuristic Algorithm

Theorem 5 means that $\hat{S} = S^*$ and $\pi^* = S^* p$ hold. Since $\mathbf{S_d}$ is finite, we can enumerate all of the elements in $\mathbf{S_d}$ and find $\hat{S} = S^*$. The size of $|\mathbf{S_d}|$ is given as m^n. Intuitively, when a strictly proper scoring rule is used, assuming that an agent acts as she declares, her reward is always better if her declaration is more "concentrated." Thus, using a non-deterministic strategy does not improve her expected utility.

In our participatory sensing example, the above S' and a truthful declaration based on it, i.e.,
$\begin{pmatrix} 0.9 \\ 0.1 \end{pmatrix}$ turn out to be optimal,
where $u(S', S'p) = 8.8 + \begin{pmatrix} 0.9 \\ 0.1 \end{pmatrix} \cdot \begin{pmatrix} 4.97 \\ 0.55 \end{pmatrix} = 13.33$.

In this example, our mechanism does not incur any efficiency loss. However, as we mentioned before, there exist cases where it might lead to efficiency loss. Here, we consider another example where the situation is identical as our participatory sensing example except we set $\alpha = 10$. In this setting, we calculated the optimal expected utility by $u(S^*, \pi^*) = 8 + \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot \begin{pmatrix} 10 \\ 0 \end{pmatrix} = 18$, where an agent's optimal declaration is done by $\pi^* = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, and optimal strategy S^* is as follows:

	sunny	cloudy	rainy
daytime	1.0	1.0	1.0
night	0	0	0

According to π^* and S^*, an agent takes pictures during the daytime regardless of tomorrow's weather. When the forecast says rain, an agent's net utility is 10, but her gross utility is 0. In this case, the amount of efficiency loss becomes 8. However, that loss is less than the maximum amount of reward 10.

Heuristic Algorithm

In this section, we propose a heuristic algorithm to find a semi-optimal strategy. Although it is possible to find S^* (as well as π^*) by exhaustively checking each element in $\mathbf{S_d}$, when m and n become too large, finding $\hat{S}$ becomes time consuming. The agent can utilize the heuristic algorithm described in Fig. 1 to find a semi-optimal strategy.

The following theorem holds.

Theorem 6. *The heuristic algorithm terminates after a finite number of iterations. Declaration $S'p$ is self-fulfilling with strategy S'.*

Proof. Assume a sequence of π, i.e., $\pi_1, \pi_2, \ldots$ and a sequence of S, i.e., $S_1, S_2, \ldots$, are obtained in the heuristic algorithm. The following inequalities hold: $u(S_1, \pi_1) < u(S_1, \pi_2) \leq u(S_2, \pi_2) < u(S_2, \pi_3) \leq u(S_3, \pi_3) < \ldots$ since $S_k p = \pi_{k+1}$ holds. Each $S_1, S_2, \ldots$, is an element of $\mathbf{S_d}$. Therefore, this sequence's length must be finite since $\mathbf{S_d}$ is a finite set.

By definition, declaring Sp and acting by S are truthful. Sp is self-fulfilling, since if there exists $j \in N$ and $s'_j \in \Pi$ such that such that $g_j \cdot s'_j + r(Sp) \cdot s'_j > g_j \cdot s_j + r(Sp) \cdot s_j$ holds, then the algorithm does not terminate at S. $\square$

Even if the agent uses the heuristic algorithm because her computational capability is limited, the principal can still obtain an accurate prediction since the obtained result remains self-fulfilling.

In our original participatory sensing example where $\alpha = 5$, if we choose initial π as $\begin{pmatrix} 0.8 \\ 0.2 \end{pmatrix}$, the algorithm chooses the first S as S'. Since $S'p \neq \pi$, it sets $\pi = S'p = \begin{pmatrix} 0.9 \\ 0.1 \end{pmatrix}$, and goes to 2. Next, it chooses the second S as S'. Since $S'p = \pi$, it terminates and returns S'.

To simplify, this paper's theoretical analysis, we assume a participant declares an entirely accurate probabilistic distribution. In practice, however, we can use a much simplified input space. For example, for several time slots, a participant chooses an answer from a small number of candidates (e.g., very likely, perhaps, unlikely).

Experimental results

In this section, we evaluate the performance of our heuristic algorithm and examine how an agent acts when her prediction of an external event includes error. We assume a situation where an unexpected scenario can happen.

Performance of approximate algorithm

First, we evaluate the performance of our heuristic algorithm. We set $n = m$ and vary n (and m) from 4 to 8. The probability that each scenario occurs is set to $1/n$. We generated 100 problem instances for each n. In each instance, we generated G by randomly selecting each $g_{i,j}$ from $[0, 100]$. We applied a spherical proper scoring rule (SPSR) as a reward function, where we set parameter α to 25, 50, and 100. Since each gross utility is at most 100, if we set $\alpha = 100$, the maximal reward value is comparable to the maximal gross utility. Initial π is obtained from a deterministic strategy that optimizes the expected utility assuming no reward is provided.

We compared the expected utility of the obtained strategies (and declarations) by the heuristic algorithm against the agent's maximal expected utility, which is obtained by enumerating all of the deterministic strategies (Fig. 2). We also examined the number of iterations of this algorithm (Fig. 3).

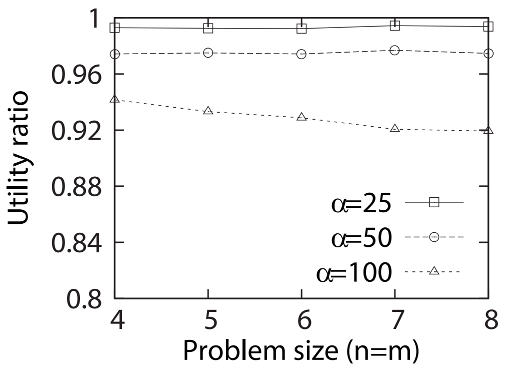

Figure 2: Utility ratio of heuristic algorithm

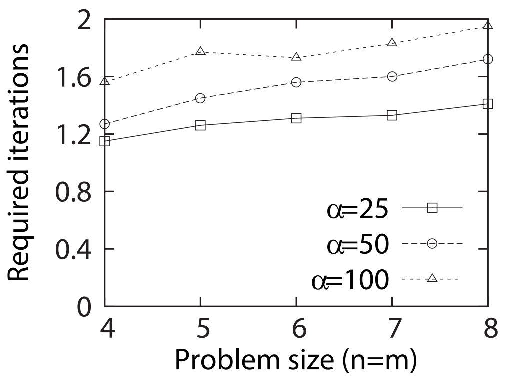

Figure 3: Required iterations of heuristic algorithm

The heuristic algorithm's expected utility always exceeds 92% of the optimal value. The heuristic algorithm is very efficient; the average number of iterations is around 2. If we increase α, the ratio becomes relatively small and the required iterations becomes relatively large. By increasing α, we assume that the optimal or semi-optimal declarations move far from the initial π used in the heuristic algorithm.

Handling unexpected scenarios

Next, we examine how an agent acts when an unexpected scenario happens. The parameter settings are basically the same as in the previous subsection, but we chose $m = 8$ and $n = 4$. Based on this knowledge, an agent obtains/declares optimal or semi-optimal declaration π. However, the next day, we assume the agent faces an unexpected scenario, where the gross utilities of each action are randomly selected from $[0, 100]$. Then, she chooses the best action to maximize her net utility.

Action i is supported in π when $\pi_i > 0$. As long as the agent chooses a supported action, we assume that the choice is acceptable for the principal, since he was prepared for it. However, if the action is not supported, he faces difficulty. A scenario is acceptable for the principal if the agent chooses a supported action. We generated 100 unexpected scenarios and calculated the ratio of acceptable scenarios (Fig. 4).

We also show the ratio of supported actions. When $\alpha = 0$,

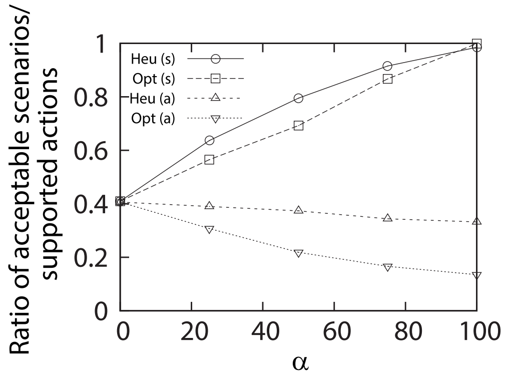

Figure 4: Ratio of acceptable scenarios ((a): action, (s): scenario)

i.e., when no reward is given, a scenario is acceptable only if the action, whose gross utility is optimal, is supported by chance. By increasing α, the agent is more likely to choose a supported action (though its gross utility is sub-optimal). As a result, the ratio of acceptable scenarios increases. On the other hand, the number of acceptable actions decreases according to the reward increases, since a proper scoring rule is convex and she can increase her utility by reducing the number of possible actions. Interestingly, the ratio of acceptable scenarios is larger when the agent uses an approximate algorithm because the ratio of the supported actions is larger.

Conclusions

We studied a mechanism to incentivize agents who predict their own future actions and truthfully declare their predictions. We addressed how a reward mechanism affects an agent's behavior and elicits a self-fulfilling declaration. To the best of our knowledge, this is the first study that shows that strictly proper scoring rules can solve to this question.

First, we proved that if a mechanism utilizes a strictly proper scoring rule, then it can elicit a self-fulfilling declaration, assuming an agent can find an optimal declaration that maximizes her expected utility and that her prediction of the external event is correct. Furthermore, we examined a method through which an agent finds an optimal declaration and proved that she can find a pair that consists of an optimal declaration and an optimal strategy by enumerating a finite set of deterministic strategies. We also developed an approximate algorithm to find a semi-optimal declaration, which is still self-fulfilling.

Future work will clarify the complexity of finding an optimal strategy. We believe that it must be NP-hard, but we have not proved it yet. We also want to extend our model to handle more complex situations, such as cases where the gross utility of one agent is affected by the actions of other agents.

Acknowledgments

This work was supported by JST PRESTO program and Japan Society for the Promotion of Science, Grant-in-Aid for Scientific Research (S) 24220003.

References

Bacon, D. F.; Chen, Y.; Kash, I.; Parkes, D. C.; Rao, M.; and Sridharan, M. 2012. Predicting your own effort. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, 695–702.

Boutilier, C. 2012. Eliciting forecasts from self-interested experts: Scoring rules for decision makers. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, 737–744.

Brier, G. W. 1950. Verification of forecasts expressed in terms of probability. *Monthly Weather Review* 78(1):1–3.

Cavallo, R., and Jain, S. 2013. Winner-take-all crowdsourcing contests with stochastic production. In *Proceedings of the 1st AAAI Conference on Human Computation and Crowdsourcing (HCOMP2013)*.

Chen, Y., and Kash, I. A. 2011. Information elicitation for decision making. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011)*.

Chen, Y., and Pennock, D. M. 2010. Designing markets for prediction. *AI Magazine* 31(4):42–52.

Chen, Y.; Gao, X. A.; Goldstein, R.; and Kash, I. A. 2011. Market manipulation with outside incentives. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI 2011)*, 614–619.

Conitzer, V. 2009. Prediction markets, mechanism design, and cooperative game theory. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI2009)*, 101–108.

Gneiting, T., and Raftery, A. E. 2007. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association* 102(477):359–378.

Law, E., and Ahn, L. V. 2011. *Human Computation*. Morgan & Claypool Publishers.

Lin, C. H.; Mausam; and Weld, D. S. 2012. Crowdsourcing control: Moving beyond multiple choice. In *Proceedings of the 4th Human Computation Workshop*, 25–26.

Matheson, J. E., and Winkler, R. L. 1976. Scoring rules for continuous probability distributions. *Management Science* 22(10):1087–1096.

Othman, A., and Sandholm, T. 2010. Decision rules and decision markets. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, 625–632.

Reddy, S.; Estrin, D.; and Srivastava, M. B. 2010. Recruitment framework for participatory sensing data collections. In *Proceedings of the 8th International Conference on Pervasive Computing*, 138–155.

Sakurai, Y.; Okimoto, T.; Oka, M.; Shinoda, M.; and Yokoo, M. 2013. Ability grouping of crowd workers via reward discrimination. In *Proceedings of the 1st AAAI Conference on Human Computation and Crowdsourcing (HCOMP2013)*.

Savage, L. J. 1971. Elicitation of personal probabilities and expectations. *Journal of the American Statistical Association* 66(336):783–801.

Shi, P.; Conitzer, V.; and Guo, M. 2009. Prediction mechanisms that do not incentivize undesirable actions. In *Proceedings of the 5th International Workshop on Internet and Networks economics (WINE09)*, 89–100.

Witkowski, J., and Parkes, D. C. 2012. A robust bayesian truth serum for small populations. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI 2012)*, 1492–1498.

Wolfers, J., and Zitzewitz, E. 2004. Prediction markets. *Journal of Economic Perspectives* 18(2):107–126.