Robust Mechanisms for Information Elicitation

Aviv Zohar Jeffrey S. Rosenschein

School of Engineering and Computer Science
The Hebrew University, Jerusalem, Israel
{avivz, jeff}@cs.huji.ac.il

Abstract. We study information elicitation mechanisms in which a principal agent attempts to elicit the private information of other agents using a carefully selected payment scheme based on proper scoring rules. Scoring rules, like many other mechanisms set in a probabilistic environment, assume that all participating agents share some common belief about the underlying probability of events. In real-life situations however, the underlying distributions are not known precisely, and small differences in beliefs of agents about these distributions may alter their behavior under the prescribed mechanism.
We propose designing elicitation mechanisms in a manner that will be robust to small changes in belief. We show how to algorithmically design such mechanisms in polynomial time using tools of stochastic programming and convex programming, and discuss implementation issues for multiagent scenarios.

1 Introduction

Game theory and decision theory tools have long been used to predict the behavior of rational agents in various settings. The common assumption made in game theory is that agents seek to maximize their own gains and do not take any other aspect of encounters into consideration. In games with an element of chance, which have an outcome that is not deterministic, a given choice can lead to many outcomes, and there are many ways to compare among actions. The most commonly used choice is to compare according to expected returns from taking the action. Other methods incorporate other measures such as risk. In any case, agents take into consideration the probability distribution for various outcomes of the game, or for the type of opponent they are facing and its selected actions.

When given the task of designing a system with interactions among agents, the designer faces a different problem. The mechanism must be designed so as to induce a rational, self-interested agent to behave in a predictable, desirable manner. The *mechanism design* literature provides many successful examples of mechanisms that "battle" the agent's self interest and successfully achieve outcomes that are more socially oriented, or are beneficial to the designing agent in some way (see [9] for a review).

In this paper we explore mechanisms for information elicitation. We assume that some principal agent wishes to buy information in a probabilistic environment, perhaps to predict some future event. In the scenario we shall explore, the information that is obtained can only be verified probabilistically. Therefore, there is often no clear-cut way to expose the seller of information as a liar. The proper incentives for truthfulness can

be obtained by carefully selecting the payments to the seller. In cases where common knowledge of probabilities of events exists, it is easy to design the payment mechanism.

However, in many real-world situations, agents cannot know the underlying probability distributions of a game, but can only assess them by sampling or other methods. Since the underlying probabilities cannot be directly known, there may be differences among agents that have access to various sources of information, or that have different priors over these distributions. The probability an agent assigns to some event is then better characterized as a *belief* it holds regarding that event.

This problem is often addressed by direct revelation mechanisms that reveal to the mechanism all needed information, including the type (or belief) of the participating agent. The mechanism then takes this information into account and acts optimally on behalf of the agent, eliminating any need to be untruthful. In information-trade settings it is unlikely that the seller of information would be interested in direct revelation. Since information is the primary commodity, revealing more of it to the mechanism is unwise, and the agent's beliefs about probabilities contain extra information.

In this sense, information elicitation scenarios are different from classical preference elicitation problems. There are other important differences: in preference elicitation scenarios, the participating agents focus less on the correctness of the information given, and more on the final outcome that is decided. This is often used by the mechanism designer; for example, in an auction scenario, an agent will not overvalue an item because it might actually be awarded an item it does not want for an inflated price. In pure information elicitation, the information the agent gives will not affect the outcome for him, but may only affect the payment it receives.

We suggest that in cases where there is some uncertainty regarding the beliefs of agents, information elicitation mechanisms must be designed without full revelation and furthermore, must be designed for robustness, not only against the manipulations players may attempt, but also for differences in the beliefs they may hold. Our contribution in this paper is to define a notion of belief-robustness in information elicitation mechanisms. We show efficient algorithms for finding robust mechanisms when such mechanisms are possible, and examine the complications that arise when attempting to extend the notion to the multiagent case.

2 Related Work and Mathematical Background

Research in artificial intelligence and on the foundations of probability theory [11] has considered probabilities as beliefs, and several models have been suggested — for example, probabilities over probabilities (which stirred much debate and controversy). [3] examined the case of agents with uncertainty about the utility functions in the world. In that case, an agent acts according to the "expected expected utility" it foresees as it takes into consideration its own uncertainty.

The idea of finding robust solutions to problems, obviously, is not new. In control theory for example, the problem of model uncertainty has been successfully tackled by implementing robust control systems that rely on feedback to correct any errors in the system's behavior [19]. Research into robust solutions was carried out in many areas. Examples include genetic algorithms [17], planning [7], and more.

In [4, 13] Conitzer and Sandholm proposed applying algorithmic mechanism design to specific scenarios as a way of tailoring the mechanism to the exact problem at hand, and thereby developing superior mechanisms. The designing algorithm is given an optimization problem — find the best possible mechanism (according to some prescribed criteria) that works for the setting at hand. They show various complexity results for the mechanism design problem. When solving the design problem, the probability distributions are considered part of the input and serve as fixed parameters for the optimization.

Other uses for information elicitation exist in multi-party computation [15, 16], where some function of the agents' secrets is computed, but agents may have reservations about revealing or computing their own secret. Another area in which information elicitation is implemented is polling. It is often of public interest to know which candidate is going to win in elections, but regular polls are notoriously inaccurate. To overcome this poor performance, a different method has been suggested for the elicitation and aggregation of information, namely the information market approach [2, 18]. There, agents are free to buy and sell options that will pay them an amount that is dependent on the outcome of some event (like some specific candidate winning an election).

2.1 Strictly Proper Scoring Rules

Scoring rules [5, 14, 6] are used in order to assess and reward a prediction given in probabilistic form. A score is given to the predicting expert that depends on the probability distribution the expert specifies, and on the actual event that is ultimately observed. For a set Ω of possible events and $\mathcal{P}$, a class of probability measures over them, a scoring rule is then defined as a function of the form: $S : \mathcal{P} \times \Omega \rightarrow \mathbb{R}$.

A scoring rule is called *strictly proper* if the predictor maximizes its expected score by saying the true probability of the event, and receives a strictly lower score for any other prediction. That is: $E_{\omega \sim p}[S(p, \omega)] \geq E_{\omega \sim p}[S(q, \omega)]$ where equality is achieved iff $p = q$. [5, 6] show a necessary and sufficient condition for a scoring rule to be strictly proper which allows easy generation of various proper scoring rules by selecting a bounded convex function over $\mathcal{P}$. Several commonly known scoring rules are the spherical scoring rule $S(p, \omega) = \frac{p_\omega}{\sqrt{\sum_{\omega'} {p_{\omega'}}^2}}$, the logarithmic scoring rule $S(p, \omega) = log(p_\omega)$, and the quadratic scoring rule $S(p, \omega) = 2p_\omega - \sum_{\omega'} p_{\omega'}^2$.

An interesting use of scoring rules within the context of a multiagent reputation system was suggested by [10], who use payments based on scoring rules to create the incentive for agents to honestly report about their experience with some service provider.

2.2 Stochastic Programming

Stochastic Programming [8] is a branch of mathematical programming where the mathematical program's variables are not precisely known. A typical stochastic program formulation consists of a mathematical program composed of constraints, a target function to optimize, and some range of allowed parameters to the problem with or without a probability distribution over them. The program is then considered in a two-step manner. The first step involves the determination of the program's variables, and in the

second step, the parameters to the problem are selected from the allowed set. The variables set in the first stage are then considered within the resulting instantiation of the problem. Therefore they must be set in a way that will be good for all (or most) possible problem instances. There are naturally several possible ways to define what constitutes a good solution to the problem. In this paper, we use the following formulation:

Given a linear programming problem of the form:

$$\begin{aligned} &\min\ c \cdot x \\ &\text{s.t.} \\ &\quad Ax \geq b \end{aligned}$$

A is considered to be from an allowed set of parameters $\mathcal{A}$. The solution x to the mathematical program must be feasible for all possible $A \in \mathcal{A}$. This form of robust optimization is presented in [1], which discusses the property of the solution and offers efficient algorithms for solving such problems. In our case, the set $\mathcal{A}$ will consist of the set of possible beliefs of agents, and the variables x will consist of the payments given to agents according to the observed outcome and what they have reported. The mathematical program will be composed of constraints to ensure truth telling, individual rationality, and effort investment on the part of the agents involved. These constraints will be solved while optimizing some function — for example, the expected payment of the mechanism designer can be minimized.

3 The Information Elicitation Scenario

The scoring rule literature usually deals with the case in which the predicting expert is allowed to give a prediction from a continuous range of probabilities, but we look at a slightly different problem: we assume each agent (including the principal agent) has access to a privately-owned random variable that takes a finite number of values only.

Modeling the knowledge of agents in the form of random variables seems natural when looking at scenarios with multiple agents, since once agents report the value of the private variable, the principal has a clear and simple way of aggregating their information with other agents' reports. The discrete possibilities also allow us more freedom in choosing our mechanism, since we do not have to induce truth-telling between infinitesimally different probabilities (as in the continuous case), and it will allow us to tailor the mechanism to the exact scenario at hand. Knowledge that is sold is very often natural to present discreetly. For example, a person acquiring weather information could be interested in the temperature forecast for the next day, but would not really care if the exact temperature is off by a single degree. The required information in this case might be given just to make a discreet choice of how warm to dress. Continuous data would then be made discrete according to the various actions that it implies.

We shall denote the principal agent's variable by Ω, and the variables of some agent i by X_i, and assume that it costs the agent c_i to access its variable and learn its precise instantiation. Since access to information is costly, the seller may have an incentive to guess at the information instead of investing the effort to learn the truth. Another possibility is that the seller will miss-report if it believes a lie would help it gain more. The

buyer of information, having only access to Ω (which may be only loosely correlated with X) might not be able to tell the difference.

The discrete values each variable may take are assumed to be common knowledge. We also assume that there is an underlying probability distribution $Pr(\Omega, X_1, \ldots, X_n)$ which (for the time being) we shall consider as known to all participants. The mechanism designer needs to decide on a payment scheme which consists of the payment to each agent i in case of an observation of ω by the principal and the reported observations $x_1, \ldots, x_n$ by the agents. We shall denote that payment by: $u^i_{\omega,x_1,\ldots,x_n}$. A payment scheme shall be considered proper if it creates the incentive for agents to enter the game, invest the effort into revealing their variable, and to tell the true value they found. These three requirements are defined more precisely below.

3.1 The Single Agent Case

For ease of exposition, we shall first look at the restricted case of a single agent (we shall return to the multiagent case later). In the case of one participating agent with a single variable, we need to satisfy three types of constraints in order to have a working mechanism. For convenience, we drop the index i of the agent and denote by $p_{\omega,x}$ the probability $Pr(\Omega = \omega, X = x)$.

1. **Truth Telling.** Once an agent knows its variable is x, it must have an incentive to tell the true value to the principal, rather than any lie x'.

$$\forall x, x' \quad s.t. \quad x \neq x' \quad \sum_{\omega} p_{\omega,x} \cdot u_{\omega,x} > \sum_{\omega} p_{\omega,x} \cdot u_{\omega,x'} \tag{1}$$

   Remember that $p_{\omega,x}$ is the probability of what actually occurs, and that the payment $u_{\omega,x'}$ is based only on what *the agent* reported.
2. **Individual Rationality.** An agent must have a positive expected utility from participating in the game:

$$\sum_{\omega,x} p_{\omega,x} \cdot u_{\omega,x} > c \tag{2}$$

3. **Investment.** The *value of information* for the agent must be greater than its cost. Any guess the agent makes without actually computing its value must be less profitable (in expectation) than paying to reveal the true value of the variable and revealing it:

$$\forall x' \quad \sum_{\omega,x} p_{\omega,x} \cdot u_{\omega,x} - c > \sum_{\omega,x} p_{\omega,x} \cdot u_{\omega,x'} \tag{3}$$

Note that all of the above constraints are linear, and can thus be applied within a linear program to minimize, for example, the expected cost of the mechanism to the principal agent: $\sum_{\omega,x} p_{\omega,x} \cdot u_{\omega,x}$.

There are naturally cases when it is impossible to satisfy the constraints. The following proposition gives a sufficient condition for infeasibility in the single agent case:

Proposition 1. *If there exist $x, x' \in X$ and $\alpha \geq 0$ such that $x \neq x' \quad \forall \omega \quad p_{\omega,x} = \alpha \cdot p_{\omega,x'}$, then there is no way to satisfy truth-telling constraints for x and x' at the same time.*

Proof. When looking at the two truthfulness constraints for x, x' we get:

$$0 < \sum_{\omega} p_{\omega,x} \cdot (u_{\omega,x} - u_{\omega,x'}) < 0 \tag{4}$$

Which is a contradiction.

We can regard this feasibility condition as a requirement of independence between the vectors $\vec{p}_x \triangleq (p_{\omega_1,x} \dots p_{\omega_k,x})$ of any two different x, x'. We shall later see that the dissimilarity between these vectors actually limits the robustness of the mechanism.

Next, we shall see that the condition described above is not only sufficient for infeasibility, but it is also necessary. Moreover, once we have a feasible solution, we can easily turn it into an optimal solution with a cost of c.

Proposition 2. *If the probability vectors $\vec{p}_x$ are pairwise independent, then there is a solution to the design problem with a mean cost as close to c as desired. This solution is optimal, due to the individual rationality constraint.*

Proof. We can easily build an optimal solution by using a strictly proper scoring rule:

$$u_{\omega,x} = \alpha \cdot S(Pr(\omega|x), \omega) + \beta_{\omega} \tag{5}$$

for some positive α, and some value β_{ω}. Since the independence relation holds for every pair x, x', the probabilities $Pr(\omega|x)$ are distinct and the scoring rule assures us of the incentive for truth telling regardless of values of α, β_{ω}.

In order to satisfy the investment constraint one can scale the payments until the value of information for the agent justifies the investment. setting:

$$\alpha > \max_{x'}[\frac{c}{\sum_{\omega,x} p_{\omega,x}(S(Pr(\omega|x), \omega) - S(Pr(\omega|x'), \omega))}] \tag{6}$$

satisfies that constraint for every x'. This is also shown in [10].

We have assumed here implicitly that the agent is risk-neutral and does not care about the risk (which may not always be a suitable assumption). Finally, we can use the β_{ω} values to satisfy the remaining individual rationality constraint tightly.

$$\beta_{\omega} = \beta > c - \alpha \sum_{\omega,x} p_{\omega,x} \cdot S(Pr(\omega|x), \omega) \tag{7}$$

We have thus shown a payment scheme with the minimal cost for every elicitation problem where different observations of X entail different probability distributions of ω. Later, we shall see that when considering robust mechanisms, the principal agent must always pay more than this.

3.2 The MultiAgent Case

When constructing a mechanism with many participating agents, we should naturally take into consideration the possible actions they are allowed to take. We will assume

here that agents cannot transfer information or utility among themselves, and must act independently. If agents do act as a single coalition without each being self-interested, they can be treated as a single agent that has access to many random variables which it may learn independently. Other behavior models that include tension among agents within a coalition, or coalition formation questions themselves, are also interesting but are beyond the scope of this work.

In the multiagent case, the mechanism designer has more freedom in creating the mechanism. As with the classic mechanism design problem, there is the option of building a dominant strategy mechanism, or solving with the weaker concept of a Nash equilibrium. It is possible to design the information elicitation mechanism to work in dominant strategies, simply by treating the variable X_i of each agent i independently and condition payments to agent i only on its variable and the outcome variable Ω. The mechanism is then designed for each agent as if it were a single-agent scenario. However, it is also possible to design the mechanism to work only as a Nash equilibrium, and condition the payments to agent i on the reports of all other agents as well. Each choice yields a different linear program that needs to be solved in order to find appropriate payments. This gives the designer further degrees of freedom with which to operate. It is easy to see that there are cases where a dominant strategy mechanism does not exist, but a mechanism that works in equilibrium does. For example, Table 1 presents such a scenario.

x_1	x_2	$Pr(x_1, x_2)$	$Pr(\omega = 1 \mid x_1, x_2)$
0	0	1/4	0
0	1	1/4	1-ϵ
1	0	1/4	1
1	1	1/4	ϵ

The elicitation scenario depicted here describes two random bits, each belonging to a different agent. The outcome bit Ω is almost the XOR of the bits of the two agents. ϵ is assumed to be positive but small.

Table 1. A Two-Agent Elicitation Scenario

A dominant strategy mechanism for agent 2 does not exist, since $Pr(\Omega = \omega | X_2 = 1) = Pr(\Omega = \omega | X_2 = 0)$ which makes it impossible to induce truth telling for the agent when conditioning the payments only on its report and on ω. Note, however, that given agent 1's report, ω is determined almost with certainty. This allows for a simple mechanism for which truth telling is a Nash equilibrium: both agents get a payment if the result matches the XOR of their reported bits, and a penalty if it does not.

A Mixture of Solution Concepts A common problem with mechanisms that work in equilibrium only, is that there may be more than one equilibrium in the game. The mechanism we have just described is no different. Consider, for example, the case where $\epsilon = 0$. The strategy of always saying the opposite of the actual result is also in equilibrium when used by both players. A possible solution is to construct a dominant strategy solution for some players, and design the solution for the other players to ensure that good behavior is the best response to the dominant strategy of the first group.

For example, we can design a mechanism for the scenario from Table 1 in the case of a positive ϵ in the following manner: agent 1's payments are conditioned only on its own reports in such a way as to induce good behavior. Such a mechanism is possible for agent 1, since the variable Ω is slightly biased to match the variable X_1. Since its payment does not depend on the actions of agent 2, agent 1 has a dominant strategy of truth telling. Agent 2's payment is then designed with the assumption that agent 1's information is known. In that case, agent 2 can rationally decide that agent 1 is going to tell the truth, and decide to do the same in order to maximize its utility.

This example can be easily generalized to a scenario with more agents. Once some order $\prec$ is imposed over the agents, the mechanism can be designed so that the payment to agent i will depend on its own report, on Ω and on any other agent j for which $j \prec i$. A mechanism designed in such a manner has only a single Nash equilibrium, and is thus more appealing. The problem is that such a mechanism may not always exist, since we are conditioning payments on less than all the available information. The order $\prec$ that is imposed on the agents is also important and different orders may certainly lead to different mechanisms. Later in this paper we shall discuss another appealing property of mechanisms constructed in this manner: they lead to finite belief hierarchies when agents need to reason about each other's unknown beliefs.

4 Building Belief-Robust Mechanisms

We shall now relax the assumption of a commonly known probability distribution, which we have used so far. We will still assume that events are governed by some probability distribution, and that agents have "close" notions of those distributions. We denote the beliefs of the mechanism designer by $\hat{p}$ and the belief of a participating agent by $p = \hat{p}+\epsilon$, where ϵ is small according to some norm. We have opted for the L_∞ norm[1] for this paper, because it is easily described using linear constraints. Other norms may also be used, and will yield convex optimization problems that are not linear. We now define the robustness level of a given solution $\vec{u}$ for some problem as follows:

Definition 1. *We shall say that a given* payment scheme $u_{\omega,x}$ *is* ϵ-robust *for an elicitation problem with distribution $\hat{p}_{\omega,x}$ if it is a proper solution to every elicitation problem with distribution $\hat{p}_{\omega,x} + \epsilon_{\omega,x}$ such that $\|\vec{\epsilon}\|_\infty < \epsilon$, and is infeasible for at least one problem instance of any larger norm.*

The definition above is very conservative, and requires feasibility for every possible difference in beliefs. Another possible approach is to give a probability over possible beliefs of the agents involved and require that the mechanism work well in a large enough portion of the cases.

Now, notice that if we are given a solution to the mechanism design problem, we can easily calculate its robustness level ϵ by solving a linear programming problem for every constraint. We do this by looking for the worst-case $\epsilon_{\omega,x}$, which stands for the worst possible belief the participating agent may hold. For example, we can write the following program to find the worst case for one of the truth-telling constraints:

[1] This norm simply takes the maximum over all coordinates.

$$
\begin{aligned}
&\min \epsilon \\
&\text{s.t.} \\
&\quad \sum_{\omega}(\hat{p}_{\omega,x} + \epsilon_{\omega,x})(u_{\omega,x} - u_{\omega,x'}) \leq 0 \\
\forall x, \omega \ &\hat{p}_{\omega,x} + \epsilon_{\omega,x} \geq 0 \\
&\sum_{\omega,x} \epsilon_{\omega,x} = 0 \\
\forall x, \omega \ &-\epsilon \leq \epsilon_{\omega,x} \leq \epsilon
\end{aligned}
$$

In the program above, only ϵ and $\epsilon_{\omega,x}$ are variables. Everything else is known and fixed. The linear problems for other constraints are easily built by substituting the first constraint above, with the negation of one of the constraints in the original design problem. Once we have solved similar linear programs for all the constraints in the original design problem, we simply take the minimal ϵ found for them as the level of robustness for the solution. The solution also provides us with a problem instance of distance ϵ for which the algorithm fails.

We can also try and find a solution with a given robustness level ϵ using the following stochastic program:

$$
\begin{aligned}
&\min \sum_{\omega,x} \hat{p}_{\omega,x} \cdot u_{\omega,x} \\
&\text{s.t.} \\
\forall x \neq x' \ &\sum_{\omega} p_{\omega,x}(u_{\omega,x} - u_{\omega,x'}) > 0 \\
&\sum_{\omega,x} p_{\omega,x} \cdot u_{\omega,x} > c \\
\forall x' \ &\sum_{\omega,x} p_{\omega,x}(u_{\omega,x} - u_{\omega,x'}) > c \\
&\text{where:} \\
\forall x, \omega \ &p_{\omega,x} = \hat{p}_{\omega,x} + \epsilon_{\omega,x} \\
&p_{\omega,x} \geq 0 \\
&\sum_{\omega,x} p_{\omega,x} = 1 \\
&-\epsilon \leq \epsilon_{\omega,x} \leq \epsilon
\end{aligned}
$$

Meaning that p is a distribution that is close to $\hat{p}$ up to ϵ, according to the L_∞ norm.

The stochastic program presented above is solvable in polynomial time, using convex programming methods. It is shown in [1] that this can be done using standard convex optimization methods, by producing a separation oracle that efficiently separates between a given point and the convex set of feasible solutions. Their separation oracle is produced by finding a problem instance from the set of possible instances which the given point clearly violates. We have already seen that this counterexample can be found in polynomial time by solving a linear program for every constraint in the original formulation, and taking the worst case among all programs.

Now that we know how to solve the problem in polynomial time, what kind of solutions can we expect to see? We have already seen that for the program instance for which $\forall \omega, x \quad \epsilon_{\omega,x} = 0$ (which corresponds to the original, non-robust design problem), the mechanism designer must pay at least c, and that a mechanism that pays infinitesimally more than c always exists (if any mechanism exists). A robust payment scheme, however, is required to cope with *any* possible belief variation Therfore, the payment

the buyer must pay will be at least as bad as that of any specific problem instance, and possibly even worse.

We can examine the robust problem formulation itself and show that here the principal must pay more than in the non-robust case. Let us assume that we were given a solution with a target function of $\gamma = \sum_{\omega,x} \hat{p}_{\omega,x} \cdot u_{\omega,x}$. Since it is not possible (due to the other constraints) that all $u_{\omega,x}$ are 0, then there exists a perturbation of beliefs $\epsilon_{\omega,x}$ which is negative for the largest $u_{\omega,x}$ and is positive for the smallest one, which then yields a strictly lower payment than γ according to the belief of a participating agent. Therefore, in order to satisfy the individual rationality constraint, γ must be strictly larger than c, which means that the mechanism designer must pay more in expectation.

Definition 2. *We define the* robustness level ϵ^* of the problem $\hat{p}$ *as the supremum of all solution robustness levels* ϵ *for which the stochastic program is solvable:*

$$\epsilon^* \triangleq \sup_{\vec{u}} \{\epsilon | \vec{u} \text{ is an } \epsilon\text{-robust solution to } \hat{p}\} \tag{8}$$

We shall now show that the robustness level of the problem in the single-agent case is determined solely by the truth-telling constraints, and is thus independent of the effort the selling agent invests in learning the information.

Proposition 3. *If a given solution* $u_{\omega,x}$ *is* ϵ*-robust with respect to the truth-telling constraints only, then it can be transformed into an* ϵ*-robust solution to the entire design problem.*

Proof. We achieve this by simply scaling the solution to give robustness for the investment constraint and shifting it to add robustness to the incentive compatibility constraint. Since the solution is ϵ-robust for the truth-telling constraints we have:

$$\forall \vec{\epsilon} \quad s.t. \quad \|\vec{\epsilon}\| < \epsilon \quad \forall x \neq x' \quad \sum_{\omega} p_{\omega,x}(u_{\omega,x} - u_{\omega,x'}) > 0. \tag{9}$$

If we sum over x we get:

$$\sum_{\omega,x} p_{\omega,x}(u_{\omega,x} - u_{\omega,x'}) > \delta_{x',\vec{\epsilon}} > 0. \tag{10}$$

Now multiplying every $u_{\omega,x}$ by a factor $\alpha = \max_{x',\vec{\epsilon}} \frac{c}{\delta_{x'\vec{\epsilon}}}$ will not hurt any of the truth-telling constraints, but will yield:

$$\forall \vec{\epsilon} \quad \forall x' \quad \sum_{\omega,x} p_{\omega,x}(u_{\omega,x} - u_{\omega,x'}) > c \tag{11}$$

which satisfies all of the investment constraints, for any possible belief change.

Next, the solution can be shifted to satisfy the individual rationality constraint, without hurting the robustness with regards to the previous constraints. We can simply add a constant β to every payment:

$$\beta > c - \min_{\omega,x}[u_{\omega,x}]. \tag{12}$$

We will thus get a solution u^* that satisfies

$$\forall \omega, x \quad u^*_{\omega,x} > c \tag{13}$$

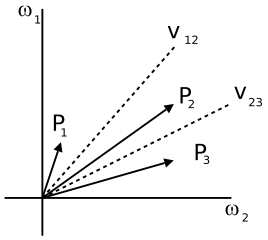
and therefore satisfies

$$\sum_{\omega,x} p_{\omega,x} \cdot u^*_{\omega,x} > \sum_{\omega,x} p_{\omega,x} \cdot c = c \tag{14}$$

for all possible belief changes, which means that u^* is ϵ-robust.

4.1 Robustness Level in 2-Dimensional Case

In cases where there are only two outcomes Ω that the principal can see, there is a simple way to find the robustness level of the problem. In this case, the vectors $\vec{p}_x$ are all two-dimensional. If a mechanism exists at all (with any robustness level) then they are pairwise independent, and therefore are spread over the first quadrant of the plane. Figure 1 depicts a scenario in two dimensions, where X takes 3 values.



The 2 axes correspond with the probabilities of the two possible results so all probability vectors are in the 2D plane.

Fig. 1. An Elicitation Scenario with 2 Possible Results

If we examine the truth telling constraints in vector form:

$$\forall x, x' \quad \vec{p}_x \cdot (\vec{u}_x - \vec{u}_{x'}) > 0 \tag{15}$$

and define $\vec{v}_{x,x'} = \vec{u}_x - \vec{u}_{x'}$, we can see that the constraints amount to finding linear separators between the vectors $\vec{p}_x$. With this geometric interpretation in mind, a robust solution is one where perturbing the vectors $\vec{p}_x$ slightly will not make them cross over to the other side of the hyper-plane defined by $\vec{v}_{x,x'}$. This amounts to finding large-margin separators between the vectors — these are separators that are of maximal distance from the points they separate.

In the two dimensional case, as vectors are spread out on the plane, it is most important to separate adjacent vectors well as these vectors are the ones closest together. The separation between non-adjacent vectors is always better. We shall not show the full proof of this here (due to lack of space), but instead only outline some propositions that lead to the proof. The next proposition will show us that in 2 dimensions, the best separating hyper-plane is the same no matter which norm we use:

Proposition 4. *If points $\vec{p}_1, \vec{p}_2 \in \mathbb{R}^2$ are separated by a plane $\vec{v}_{1,2}$ for which $\vec{v}_{1,2} \cdot (\vec{p}_1 + \vec{p}_2) = 0$, then the distance of both points from the separating plane is equal under any metric L_i.*

Now, without loss of generality, we can assume the vectors $\vec{p}_1 \dots \vec{p}_n$ are ordered in a clockwise direction. We can then find $n-1$ vectors of the form $\vec{v}_{i,i+1}$ which are linear separators between $i, i+1$. We choose them to be the best linear separators possible between the two points. This is achieved by a vector for which $\vec{v}_{i,i+1} \cdot (\vec{p}_i + \vec{p}_{i+1}) = 0$. In two dimensions, this condition determines $\vec{v}_{i,i+1}$ up to a scaling factor that does not change the separating plane.

The truth-telling constraints for $x = i, \quad x' = i+1$ can now be satisfied by setting the payments to satisfy $\vec{v}_{i,i+1} = (\vec{u}_i - \vec{u}_{i+1})$. This can easily be done by arbitrarily setting $\vec{u}_1 = \vec{0}$, and then proceeding to set the remaining vectors in sequence, where $\vec{u}_{i+1}$ is set according to the equation with $\vec{v}_{i,i+1}$. Now that we have set all the payments, we also have to show that this scheme satisfies all the other truth-telling constraints, for non-consecutive values of x, x', and that it satisfies them in a robust way. The relationship between separating planes is given by $\vec{v}_{i,j} = -\vec{v}_{j,i}$ and $\vec{v}_{i,j} = \vec{v}_{i,k} + \vec{v}_{k,j}$.

The following proposition leads to the conclusion that setting the payments only according to consecutive points works for the other points as well:

Proposition 5. *If for all i the separator defined by $\vec{v}_{i,i+1}$ separates $\vec{p}_i$ from $\vec{p}_{i+1}$ correctly then the distance of point $\vec{p}_i$ from the separator $\vec{v}_{i,i+1}$ is smaller than its distance from the separator $\vec{v}_{i,j}$ for all $j > i+1$.*

Similarly, the distance of point $\vec{p}_i$ from $\vec{v}_{i,i-1}$ is smaller than its distance from $\vec{v}_{i,j}$ for all $j < i-1$.

We have therefore seen that the actual limitation on the robustness of the mechanism is the distance between consecutive vectors p_i and p_{i+1} (or more precisely, the distance between them and the optimal separating plane between them) in the two-dimensional case. Taking the best separators between them leads to the most robust feasible solution for all of the truth-telling constraints, which in turn can be transformed into a feasible solution to the full robust design problem.

4.2 Robust Mechanisms for Multiple Agents

Designing robust mechanisms for multiple agents is a far more complex issue. The designer must now take into account not only the possible beliefs of agents about the probabilities of events, but also their beliefs about the beliefs of other agents. This is especially true when constructing a mechanism that will work only at an equilibrium. In order for an agent to believe that some strategy is in equilibrium, it must also be convinced that its counterparts believe that their strategies are in equilibrium, or are otherwise optimal. This will only occur if the agent believes that they believe that it believes that its strategy is in equilibrium — and so on to infinity.

Any uncertainty about the beliefs of other agents grows with every step up the belief hierarchy. If agent A knows that all agents have some radius ϵ of uncertainty in beliefs, and its view of the world consists of some probability distribution p it assigns to events,

then it is possible that agent B believes the distribution is p' and further believes that agent A believes the distribution is some p'' which is at a distance of up to 2ϵ from p. With an infinite belief hierarchy, it is therefore possible to reach any probability if we go high enough in the hierarchy.

A possible solution to this problem is to use the mixture of solution concepts we have seen before. If each agent's payment only depends on the actions of agents before it according to some order $\prec$, then it only needs to take their beliefs into consideration when deciding on a strategy. The necessary belief hierarchy is then finite, which limits the possible range of beliefs about beliefs. The most extreme case of this is to design the mechanism for dominant strategies only. Naturally, a solution constructed in such a way may be less efficient or may not exist at all.

One may alternatively consider bounded rational agents that are only capable of looking some finite distance into the hierarchy as possible subjects for the mechanism design. An extreme example would be agents that believe that everyone else shares their basic belief about the world, and do not reason about the beliefs of others at all (but may, in fact, have different beliefs).

It is also interesting to note, that when agents have some common shared model of the world, and with it some common prior over the possible beliefs of one another, then as [12] shows, the iterated expectations each of them considers in its infinite belief hierarchy converge to the same value. This hints at the fact that *in expectation* the beliefs in the hierarchy drift towards the shared prior.

5 Conclusions and Future Work

We have discussed discrete information elicitation mechanisms and have shown that such mechanisms can be efficiently designed to be robust to a wide range of beliefs held by the participating agents. The robust mechanisms are naturally more expensive than their non-robust counterparts. We also discussed some of the complications arising from designing the mechanism for multiple participants, and have shown some cases under which these complications can be handled easily. Further exploration in that direction, and especially an attempt to cope with the infinite belief hierarchies implied by the Nash equilibrium concept, is still required. It would also be interesting to try and build other belief-robust mechanisms, perhaps in the setting of preference elicitation.

Another interesting direction to explore is the area of collusion among agents. If agents share information and payments among themselves, it is going to be harder to design working mechanisms. Here, there are several levels of cooperation possible for the agents, ranging from only helping other agents if there is personal gain in it, to helping other agents in case it is beneficial to the coalition as a whole. Exploring the information elicitation problem from a coalition formation point of view would also be interesting, as it can be expected that as agents reveal the values of their variables, the coalitions they would want to join (in order to manipulate the mechanism) may change depending on the result.

We have used tools of stochastic programming to solve for robust solutions, but have only scratched the surface of potential uses of these tools. Other alternative problem formulations can be explored, especially formulations that include more detailed

information about the possible beliefs of agents. These would fit quite well into the mainstream work done in stochastic programming.

Finally, it would be interesting to explore the area of partially-effective mechanisms. These may fail to induce truth telling by agents in some cases, and only work well with some probability. One might explore the tradeoff between the confidence level of the designer in the mechanism, and its robustness and cost.

References

1. A. Ben-Tal and A. Nemirovski. Robust solutions of uncertain linear programs. *Operations Research Letters*, 25:1–13, 1999.
2. P. Bohm and J. Sonnegard. Political stock markets and unreliable polls. *Scandinavian Journal of Economics*, 101(2):205, June 1999.
3. C. Boutilier. On the foundations of expected expected utility. In *IJCAI-03*, pages 285–290, Acapulco, 2003.
4. V. Conitzer and T. Sandholm. Complexity of mechanism design. In *UAI-2002*, Edmonton, Canada, August 2002.
5. T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. Technical Report 463, Department of Statistics, University of Washington, 2004.
6. A. D. Hendrickson and R. J. Buehler. Proper scores for probability forecasters. *Annals of Mathematical Statistics*, 42:1916–1921, 1971.
7. T.C. Hu, A.B. Kahng, and G. Robins. Optimal robust path planning in general environments. *IEEE Transactions on Robotics and Automation*, 9(6):775–784, 1993.
8. P. Kall and S. W. Wallace. *Stochastic Programming*. Wiley-Interscience Series in Systems and Optimization. John Wiley, January 1995.
9. E. Maskin and T. Sjstrm. Implementation theory. Working paper, Harvard University and Penn State, January 2001.
10. N. Miller, P. Resnick, and R. Zeckhauser. Eliciting honest feedback: The peer prediction method. *Management Science*, 2005. Forthcoming. www.si.umich.edu/ presnick/papers/elicit/.
11. J. Pearl. *Probabilistic reasoning in intelligent systems : networks of plausible inference*. Morgan Kaufmann, San Mateo, CA, 2nd edition edition, 1988.
12. D. Samet. Iterated expectations and common priors. *Games and Economic Behavior*, 24(1), 1998.
13. T. Sandholm. Automated mechanism design: A new application area for search algorithms. In *Proceedings of the International Conference on Principles and Practice of Constraint Programming*, 2003.
14. L. J. Savage. Elicitation of personal probabilities and expectations. *Journal of the American Statistical Association*, 66(336):783–801, December 1971.
15. R. Smorodinsky and M. Tennenholtz. Sequential information elicitation in multi-agent systems. In *UAI-2004*, 2004.
16. R. Smorodinsky and M. Tennenholtz. Overcoming free-riding in multi-party computation: the anonymous case. *Games and Economic Behavior*, 2005. Forthcoming.
17. S. Tsutsui, A. Ghosh, and Y. Fujimoto. A robust solution searching scheme in genetic search. In *PPSN IV: Proceedings of the 4th International Conference on Parallel Problem Solving from Nature*, pages 543–552, London, UK, 1996. Springer-Verlag.
18. J. Wolfers and E. Zitzewitz. Prediction markets. Working Paper 10504, National Bureau of Economic Research, 2004.
19. K. Zhou, J. C. Doyle, and K. Glover. *Robust and optimal control*, chapter 9. Prentice-Hall, 1996.