Robust Mechanisms for Information Elicitation

Aviv Zohar and **Jeffrey S. Rosenschein**
School of Engineering and Computer Science
The Hebrew University of Jerusalem
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 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.

Introduction

Game theory and decision theory tools have long been used to predict or prescribe the behavior of rational agents in various settings. The common assumption made is that agents seek to maximize their own gains, but in games with an element of chance or multiple players, a given choice can lead to many outcomes, and there are various ways to compare among actions. The most common approach is to consider expected returns from taking each action (other methods incorporate measures such as attitude towards 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 complementary 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 such mechanisms (see (Maskin & Sjöström 2001) for a review).

Our Scenario

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 explore, the information that is obtained can only be verified probabilistically; thus, there is often no clear-cut way to expose the seller of information as a liar. Furthermore, we assume that the sellers and buyers are engaged in a one-shot interaction. This means that there is no record of the past reliability of the seller, on which the buyer can rely; nor will there be any future interaction through which the buyer can reward or punish the seller. The proper incentives for truthfulness can be put into place by carefully selecting the payments to the seller. If the probabilities of events are common knowledge, it is computationally easy to design the payment mechanism (discussed below).

However, in many real-world situations, agents cannot know the underlying probability distributions, but can only assess them by sampling or other methods. If the underlying probabilities cannot be directly known, there may be differences among agents with access to various sources of information, or with different priors over these distributions.

Partial Revelation for Information Elicitation

This problem is often addressed by direct revelation mechanisms that require agents to divulge all needed information, including their probability beliefs (i.e., *type*). The mechanism then takes this information into account and acts optimally on behalf of the agent, eliminating any need to be untruthful. However, in settings where information is sold, it is unlikely that the seller would be willing to participate in direct revelation schemes. 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.[1] In this sense, information elicitation scenarios are different from classical preference elicitation problems.

There is another important difference. In preference elicitation scenarios, information revelation is most often used as a means to an end (i.e., to arrive at some desirable outcome). In pure information elicitation, the information being revealed *is* the point of the transaction. Furthermore, the seller is only concerned with its payment, not any other consequence of providing one piece of information or another.

[1] It remains unwise even if the mechanism is handled by a trusted third party, since revealing extra information would be reflected in payments made by the buyer.

Contribution of the Paper

It is now commonplace, via the Internet, for information exchange to occur among strangers, with no assurance of reliability. Examples might include web services created by individuals, providing various specific types of information (e.g., weather or traffic information). A good mechanism can motivate a seller of information to be truthful, but the effectiveness of such a mechanism depends on the beliefs of the agents involved.

We suggest that in cases where there is some uncertainty regarding the beliefs of agents, mechanisms should be designed for robustness, not only against the manipulations players may attempt, but also to deal with 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.

The rest of the paper is organized as follows. In the next section we present the mathematical background to our approach. We then formally define the information elicitation scenario, and discuss the solution to the simpler case where the probabilities are all common knowledge, including various solution concepts in the multiagent case. In the section on Belief Robust Mechanisms, we relax the assumption of common knowledge, and define the notion of mechanism robustness (which deals with variations in beliefs about probabilities). We show how to optimally design mechanisms in this setting, and discuss issues of the multiagent case. We conclude with an overall discussion of related work, our approach, and of future work.

Mathematical Background

Strictly Proper Scoring Rules Scoring rules (Savage 1971) 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 $p = q$. (Gneiting & Raftery 2004; Hendrickson & Buehler 1971) 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 $\mathcal{P}$.

An interesting use of scoring rules within the context of a multiagent reputation system was suggested by (Miller, Resnick, & Zeckhauser 2005), who use payments based on scoring rules to create the incentive for agents to honestly report about their experience with some service provider.

Stochastic Programming Stochastic Programming (Kall & Wallace 1995) is a branch of mathematical programming where the mathematical program's constraints and target function are not precisely known. A typical stochastic program formulation consists of a set of parameterized constraints over variables, and a target function to optimize. The program is then considered in two phases. The first phase involves the determination of the program's variables, and in the second phase, the parameters to the problem are randomly 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 conservative formulation of (Ben-Tal & Nemirovski 1999) which requires the assignment of variables to satisfy the constraints of the program for *every* possible program instance. Linear stochastic programs such as these are efficiently solvable using convex programming tools.

In our case, each instance will correspond to a different variation in the beliefs held by the participating agents.

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. The discrete values allow us to tailor the mechanism to the exact scenario at hand without the need to differentiate between infinitesimally different cases. Knowledge that is sold is very often natural to present discretely.[2] Finally, aggregating information from several agents is also much clearer and simpler to do with discrete variables.

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 that 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 misreport if it believes a lie would increase the payment it receives. The buyer of information, having access only 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 buyer, and the observations $x_1, \ldots, x_n$ reported by the agents. We shall denote

[2] 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 one degree. The required information in this case might be given just to make a discrete choice of how warmly to dress. Continuous data can sometimes be made discrete according to the various actions it implies.

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 acquiring their variable, and to tell the true value that they found. These three requirements are defined more precisely below.

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). We assume here that the agent is risk-neutral. 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'.

$$\forall x, x' \quad s.t. \quad x \neq x' \quad \sum_{\omega} p_{\omega,x} \cdot (u_{\omega,x} - u_{\omega,x'}) > 0 \quad (1)$$

   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:

$$\sum_{\omega,x} p_{\omega,x} \cdot u_{\omega,x} > c \quad (2)$$

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

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

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 $\alpha \geq 0$ s.t. $x \neq x' \quad \forall \omega \quad p_{\omega,x} = \alpha \cdot p_{\omega,x'}$, then there is no way to satisfy truth-telling constraints for x and x' at the same time.*

Proof:

When looking at the two truth-telling constraints for x, x' we get: $0 < \sum_{\omega} p_{\omega,x} \cdot (u_{\omega,x} - u_{\omega,x'}) < 0$ which is a contradiction. □

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

Next, we shall see that if the condition described in proposition 1 does not hold, we can alway construct a proper payment scheme. Moreover, once we have some working payment scheme, we can easily turn it into an optimal one with a cost of c.

Proposition 2. *If the probability vectors $\vec{p}_x$ are pairwise independent, i.e., $\forall x, x'$ there is no λ such that $\vec{p}_x = \lambda \cdot \vec{p}_{x'}$, then there is a proper payment scheme 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:

$$u_{\omega,x} = \alpha \cdot S(Pr(\omega|x), \omega) + \beta_\omega \quad (4)$$

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.

To satisfy the investment constraint, one can scale the payments until the value of information for the agent justifies the investment. Setting

$$\alpha > \max_{x'}\left[\frac{c}{\sum_{\omega,x} p_{\omega,x}(S(Pr(\omega|x),\omega) - S(Pr(\omega|x'),\omega))}\right] \quad (5)$$

satisfies that constraint for every x'. This is also shown in (Miller, Resnick, & Zeckhauser 2005).

Finally, we can use the β_ω values to satisfy the remaining individual rationality constraint tightly:

$$\beta_\omega = \beta > c - \alpha \sum_{\omega,x} p_{\omega,x} \cdot S(Pr(\omega|x), \omega) \quad (6)$$

□

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.

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.

In the multiagent case, the mechanism designer has more freedom in creating the mechanism. There is the option of building a dominant strategy mechanism (where rational action choice of an agent does not depend on the action of any other agent), or solving with the weaker concept of a Nash equilibrium (where a given set of actions is only optimal if none of the agents deviates). 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 in equilibrium, by conditioning 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. There are cases where a dominant strategy mechanism does not exist, but a mechanism that works in equilibrium does. The following table presents such a scenario for two agents.

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-δ
1	0	1/4	1
1	1	1/4	δ

The elicitation scenario depicted here describes two random bits, each belonging to a different agent. The principal's variable Ω is almost the XOR of the bits of the two agents. δ is assumed to be positive but small.

A dominant strategy mechanism for agent 2 does not exist according to Proposition 1, 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. However, 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 $\delta = 0$. 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 in the previous table for the case of a positive δ 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. 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 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 $j \prec i$. Such a mechanism 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 one another's unknown beliefs.

Building Belief-Robust Mechanisms

We shall now relax the assumption of a commonly known probability distribution, which we have used so far. We will instead assume that agents have "close" notions of the governing probability 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[3] 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.

The Robustness Level of a Payment Scheme

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 payment scheme with regard to every elicitation problem with distribution* $\hat{p}_{\omega,x} + \epsilon_{\omega,x}$ *such that* $\|\vec{\epsilon}\|_\infty < \epsilon$, *and is not proper 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.

Determining the Robustness Level of a Mechanism We can calculate the robustness level ϵ of a given mechanism 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 that 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:

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

In the program above, only ϵ and $\epsilon_{\omega,x}$ are variables. 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 take the minimal ϵ found for them as the level of robustness for the mechanism. The solution also provides us with a problem instance of distance ϵ for which the mechanism would fail.

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

Finding a Mechanism With a Given Robustness Level We can try and find a payment scheme with a given robustness level ϵ using the following stochastic program:

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

The program considers all distributions p that are close to $\hat{p}$ up to ϵ, according to the L_∞ norm. It is solvable in polynomial time using convex programming methods, as shown in (Ben-Tal & Nemirovski 1999).

The Cost of Robust Mechanisms 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), a payment scheme that costs only infinitesimally more than c always exists (if any mechanism exists). A robust payment scheme, however, is required to cope with *any* possible belief variation and will cost more to implement.

Consider a mechanism with an expected cost 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, and the buyer must pay more in expectation.

The Robustness Level of an Elicitation Problem

We shall often be interested in finding the most robust mechanism possible for a given scenario. We therefore define the robustness level of the *problem* in the following manner:

Definition 2. *The* robustness level ϵ^* of the problem $\hat{p}$ *is the supremum of all robustness levels ϵ for which a proper mechanism exists:*

$$\epsilon^* \triangleq \sup_{\vec{u}} \{\epsilon | \vec{u} \text{ is an } \epsilon\text{-robust payment scheme for } \hat{p}\}.$$

To find the robustness level of a problem, one can perform a binary search; the robustness level is certainly somewhere between 0 and 1. One may test at every desired level in between to see if there exists a mechanism with some specified robustness by solving the stochastic program above. The space between the upper and lower bounds is then narrowed according to the answer that was received.

As in the non-robust case, the design of a robust single-agent mechanism relies only on the truth-telling constraints:

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 problem.*

Proof: [sketch] We achieve this by scaling and shifting the solution to add robustness to the other constraints in a manner similar to equations 5 and 6. In this case the payments are transformed to satisfy the requirements for every possible $\vec{\epsilon}$ with a small enough norm. □

A Bound for Problem-Robustness A simple bound for robustness of the problem can be derived from examining the truth-telling conditions. In fact, Proposition 1 for non-robust mechanisms can be viewed as a specific case of the following proposition for 0-robust mechanisms:

Proposition 4. *The robustness level ϵ^* of a problem $\hat{p}$ can be bounded by the smallest distance between a vector $\hat{p}_x$ and the optimal hyperplane that separates it from $\hat{p}_{x'}$:*

$$\epsilon^* \leq \min_{x,x'} ||\hat{p}_x - (\hat{p}_x^{tr} \cdot \vec{\varphi}_{x,x'}) \cdot \vec{\varphi}_{x,x'}||_\infty$$

$$\vec{\varphi}_{x,x'} = \frac{\hat{p}_x + \hat{p}_{x'}}{||\hat{p}_x + \hat{p}_{x'}||_2}$$

The optimal separating hyperplane is a hyperplane that separates the points and is of maximal (and equal) distance from both of them.

Proof: [sketch] If there exist x, x' that give distance ϵ to the hyperplane then the vectors $\hat{p}_x, \hat{p}_{x'}$ can be perturbed towards the hyperplane with a perturbation of norm ϵ, until they are linearly dependent. For this problem instance, according to Proposition 1, there is no possible mechanism. □

In the case where $|\Omega| = 2$, the vectors $\hat{p}_x$ are situated in a two-dimensional plane, and it can be shown that the bound given above is tight — the problem robustness is determined exactly by the closest pair of vectors.

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. 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.

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 own belief is some probability distribution p, 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).

Related Work

Research in artificial intelligence and on the foundations of probability theory has considered probabilities as beliefs,[4] and several models have been suggested — for example, probabilities over probabilities (Pearl 1988). Cases where agents have uncertainty about the utility functions in the world were examined in (Boutilier 2003); an agent acts according to the "expected expected utility" it foresees as it takes into consideration its own uncertainty.

(Conitzer & Sandholm 2002) proposed applying automated mechanism design to specific scenarios as a way of tailoring the mechanism to the exact problem at hand, and thereby developing superior mechanisms.

Other uses for information elicitation exist in reputation systems (Miller, Resnick, & Zeckhauser 2005), and in multi-party computation (Smorodinsky & Tennenholtz 2005), where some function of the agents' secrets is computed, but agents may have reservations about revealing or computing their own secret. Yet another area in which information elicitation is implemented is polling. The information market (Bohm & Sonnegard 1999; Wolfers & Zitzewitz 2004) approach has been suggested as a way to get more reliable results than regular polls. There, agents 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).

Conclusions and Future Work

We have discussed discrete information elicitation mechanisms and have shown that such mechanisms can be efficiently designed to be robust with regard 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 into 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 transfer payments, 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 doing so, to helping other agents when 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.

[4] Leading to controversy between Bayesians and Frequentists.

Acknowledgment

This work was partially supported by grant #039-7582 from the Israel Science Foundation.

References

Ben-Tal, A., and Nemirovski, A. 1999. Robust solutions of uncertain linear programs. *Op. Research Letters* 25:1–13.

Bohm, P., and Sonnegard, J. 1999. Political stock markets and unreliable polls. *Scandinavian J. of Econ.* 101(2):205.

Boutilier, C. 2003. On the foundations of expected expected utility. In *IJCAI-03*, 285–290.

Conitzer, V., and Sandholm, T. 2002. Complexity of mechanism design. In *UAI-2002*, 103–110.

Gneiting, T., and Raftery, A. E. 2004. Strictly proper scoring rules, prediction, and estimation. Technical Report 463, Department of Statistics, University of Washington.

Hendrickson, A. D., and Buehler, R. J. 1971. Proper scores for probability forecasters. *Annals of Mathematical Statistics* 42:1916–1921.

Kall, P., and Wallace, S. W. 1995. *Stochastic Programming*. Systems and Optimization. John Wiley.

Maskin, E., and Sjöström, T. 2001. Implementation theory. Working paper, Harvard University and Penn State.

Miller, N.; Resnick, P.; and Zeckhauser, R. 2005. Eliciting honest feedback: The peer prediction method. *Management Science*. Forthcoming.

Pearl, J. 1988. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. San Mateo, CA: Morgan Kaufmann, 2nd edition.

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

Smorodinsky, R., and Tennenholtz, M. 2005. Overcoming free-riding in multi-party computation: the anonymous case. *Games and Economic Behavior*. Forthcoming.

Wolfers, J., and Zitzewitz, E. 2004. Prediction markets. Working Paper 10504, National Bureau of Econ. Research.