Mechanisms for Information Elicitation

Aviv Zohar[*] Jeffrey S. Rosenschein[†]

Abstract

We study the computational aspects of information elicitation mechanisms in which a principal 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 examine two related models for the problem. The first model assumes that agents have a similar notion of the probabilities of events, and we show that this approach leads to efficient design algorithms that produce mechanisms which are robust to small changes in the beliefs of agents.

In the second model we provide the designer with a more precise and discrete set of alternative beliefs that the seller of information may hold. We show that construction of an optimal mechanism in that case is a computationally hard problem, which is even hard to approximate up to any constant. For this model, we provide two very different exponential-time algorithms for the design problem that have different asymptotic running times. Each algorithm has a different set of cases for which it is most suitable. Finally, we examine elicitation mechanisms that elicit the confidence rating of the seller regarding its information.

1 Introduction

The old aphorism "Knowledge is power", stated by Sir Francis Bacon some four centuries ago, is more relevant now than ever. The need to make informed choices causes correct and accurate information to be a desired and highly-valued commodity. As intelligent automated agents take on more tasks, and need to act independently within large systems, their need to buy and sell information increases.

[*] School of Engineering and Computer Science, The Hebrew University of Jerusalem, Jerusalem, Israel. Email: avivz@cs.huji.ac.il

[†] School of Engineering and Computer Science, The Hebrew University of Jerusalem, Jerusalem, Israel. Email: jeff@cs.huji.ac.il

Information in stochastic environments is hard to evaluate, and may be easily faked. Any novice can give a prediction regarding the behavior of tomorrow's stock market; by pure chance, those predictions may outperform those of even the most informed financial wizard.

The question that naturally arises is how to pay for information that can only be verified with some probability. This is especially important in cases where in order to obtain the information, the seller itself has to invest some effort. The payments made by the buyer must be carefully set so as to induce the seller to invest the effort into acquiring the true information. Otherwise, the seller might be tempted to avoid the cost of obtaining the information, and simply make something up.

Most current real-world information trading is done with reliable sources of information over an extended period of time (for example, buying the same newspaper every day). This repeated form of interaction helps motivate the provider of information to supply accurate and reliable reports (not unlike the "shadow of the future" motivating cooperation in the iterated Prisoner's Dilemma [1]). The potential for additional interactions in the future makes the information provider's reputation valuable, and motivates the seller to provide accurate pieces of information.

However, advances in technology and infrastructure such as the internet have made a multitude of information sources readily available at a moment's notice (via web services [2], for example). These tend to be smaller and much more specialized information providers, which can accurately report about a small niche in which they specialize. Interactions with these sources are often not repeated. Since there is no central authority that governs these sources, and no single authority can vouch for the reliability of the information they provide, it is left up to the buyer of information to sift through the information that is available and decide what to use.[1]

One approach to the problem of source reliability is the use of reputation systems [3]. These systems are mechanisms through which agents provide feedback about the quality of service they received from a specific vendor; this feedback is later viewed by other potential clients. Unfortunately, solid non-manipulable reputation systems are hard to create, and most service providers on the internet are not currently rated by any such system.

We are therefore interested in other ways of obtaining correct information from a previously unknown information source. We will assume that there is no repeated interaction, and the incentive for providing good service must exist within every transaction, on its own. The overall approach we take in this work is that of mechanism design. We shall attempt to create the incentives for delivering accurate reports by providing payments to the agents in a way that will guarantee them a higher payment when they are behaving well, i.e., when they provide correct information.

[1] As an example, consider querying some foreign weather service before traveling abroad. One will only know if the weather prediction they supplied is good after arriving at the destination. One may not be likely to require the services of that supplier again.

We shall assume that agents are acting rationally and that they are not intentionally trying to sabotage the buyer—any use the buyer may make of the purchased information does not affect the seller. Instead, we adopt the assumption that information providers are only interested in receiving a higher payment and doing the least amount of work. A truly malicious agent that is trying to intentionally deceive, regardless of monetary loss, will not give good information regardless of the mechanism applied, and must therefore be dealt with in other ways. Such agents are often handled using security and encryption tools that we shall not discuss here.

1.1 An Example Scenario for Information Elicitation

There are many possible scenarios for information exchange, such as reviewing papers, obtaining predictions about the stock market, buying weather information, and so on. We present here one example to which we will refer throughout the paper.

Let us assume that Bob owns a car, and wants to decide if he should upgrade his emergency road service coverage. For this purpose, he wants to evaluate the mechanical condition of the car; this will help him predict the car's chances of breaking down in the near future, and will help him decide whether the extra insurance is worthwhile. Since Bob knows very little about cars, he turns to an independent expert, a mechanic named Alice, and asks her to take a look under the hood.

Knowing that Bob is not an expert, Alice can decide not to invest any effort in checking the car, and instead make up some list of malfunctions that threaten to disable the car at any moment, or alternatively she may just say that the car is fine (she has no vested interest in whether Bob upgrades his coverage). How will Bob know that he was told the truth? Even if Alice invests effort in checking the car and says that the car is fine, an accidental malfunction could disable it the next day (probably making Bob feel cheated).

To ensure trust, Alice can make her wages conditioned on the future: if Alice says the car is in poor shape, Bob will get a refund if his car does not break down within the next six months, while if Alice reports that the car is fine, Bob gets a refund if the car does break down within six months. What are the exact payments that will ensure that Alice does her job? There is naturally some probability that Alice will have to refund some of Bob's money even if she checked the car and reported the truth to Bob.

There might also be a situation in which Alice knowingly lies to Bob. If the chances that a car in good condition will break down are too high, Alice could decide to say the car is in bad condition, and thus ensure that she does not refund Bob if his car breaks down (even though it was indeed in good condition).

1.2 Information Elicitation vs. Preference Elicitation

Mechanism design [4, 5] is the study of how to set the rules and protocols of interaction among agents in a way that will encourage rational agents to behave in a prescribed way that leads to a desired outcome. 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.

Many times, in order to decide on an outcome, a mechanism tries to elicit the preferences of participating agents. Information elicitation scenarios are slightly different from preference elicitation as it is usually understood in the mechanism design literature. In preference elicitation scenarios, information revelation is most often used as a means to an end (i.e., to arrive at some desirable outcome). For example, an auctioneer may want to know the valuations potential buyers have for an expensive painting so that he can award this painting to the bidder that values it highest, and in the process make more money.

In pure information elicitation, the information being revealed *is* the point of the transaction. The seller is assumed to only be concerned with its payment, not any other consequence of providing one piece of information or another. In this sense, information elicitation can be seen as a subproblem of mechanism design, where the mechanism has no outcome to determine.[2] This limitation leaves the mechanism with fewer degrees of freedom.

Since information elicitation scenarios are all about trading information, it may be important to the participants not to give out any information for free, and to keep all their extra knowledge about the world secret. Later in this article we shall examine scenarios where the seller and buyer of information possess different beliefs about the world. In classic mechanism design, 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,[3] and the agent's beliefs about probabilities contain extra information.

1.3 Contribution

As computers take on more tasks that require intelligent decisions and reliable information, and as micro-transactions of information begin to play a larger role in information trade, establishing the proper incentives for truthfulness becomes increasingly important. We present here a model for one-shot transactions of information that can incorporate these incentives, and show how to extend it in four ways:

1. We show that some mechanisms are more robust to varying beliefs of agents than other mechanisms. The notion of belief robustness that we

[2] This is similar to the classification of elections as mechanisms where no money changes hands, and only an outcome is selected.

[3] 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.

define is applicable to many real-world situations where there is no common knowledge between the seller and buyer of information, but effective mechanisms can still be constructed. We present efficient algorithms for finding such mechanisms.

2. We show that if the selling agent has additional knowledge about the state of the world that it is not willing to sell, the design of an optimal mechanism becomes computationally hard. We present two exponential-time algorithms for the design problem that exploit different aspects in the structure of the problem, and achieve different running time profiles.

3. We look at the case where the expert selling information has some uncertainty regarding its quality, and show how the confidence rating of the seller can sometimes be elicited along with the information itself.

1.4 Structure of the paper

In the next section we review related work and give a brief overview of some mathematical and computational background used in the rest of the paper. In Section 3 we define the basic information elicitation model and explore its basic properties in the case of one seller. Section 4 then explores a model where agents do not hold a common view of the world and need to design mechanisms that are robust against small differences in beliefs. Next, we turn to a scenario where the seller possesses more knowledge about the probabilities than the buyer does, and show that designing good mechanisms in this case is often hard. We give two different algorithms to design such mechanisms that have different running times. In Section 6 we discuss elicitation of confidence ratings in scenarios where the seller has some uncertainty about the quality of its information. We present our conclusions in Section 7.

2 Background

2.1 Related Work

The economic theory of contracts has dealt with principal-agent models in various forms. Some of the canonical families of models in this field include those with *adverse selection*—where an agent is asked to reveal private information about his type but may give false information. The proper incentives for truth-telling are set by a well-designed contract. Another well-known family of models thoroughly explored in economics is that of *moral hazard*. In these settings, the agents are asked to take a hidden action (one that is costly for them) that is not directly visible to the principal. Instead, the principal observes some noisy signal that is affected by the action. The aim of the contract is then to make sure that the agents gain more by performing the action. The reader is urged to refer to [6, 7] for a comprehensive introduction to these models and several of their variants. Our own work includes elements from both families—hidden action to learn the information the buyer is interested in acquiring, and truthfulness in revealing it. However, we no longer assume that probabilities of events are common knowledge, but instead treat them as beliefs held by the agents. This leads among other things to interesting computational questions that are not often explored within the economic context.

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 [8]. Cases where agents have uncertainty about the utility functions in the world were examined in [9]. There, an agent acts according to the "expected expected utility" it foresees as it takes into consideration its own uncertainty. The truthful elicitation of such beliefs has also attracted great interest [10, 11, 12] (see Section 2.2 on scoring rules).

The issue of common knowledge and common priors has been studied within the context of probability theory [13, 14]. Here, the beliefs about beliefs of agents also play a large role.

There are many natural uses for information elicitation in computer science. For example, in reputation systems [15, 16] information is elicited from agents about their experience with some service provider. This information is important for agents that will interact with that service provider in the future, but the reporting agent that has already completed the interaction needs to be motivated in some other way to reveal the results of its own interaction.

In multi-party computation settings, information is elicited in order to compute some function of the agents' secrets. Agents are interested in the correct result of the computation but do not wish to reveal their secret [17], and use cryptographic tools to conceal it. Multi-party computation scenarios where agents have to invest effort to discover their secrets have also been explored [18, 19]. In our work we do not assume that agents have reservations about revealing their secret, only that they wish to maximize their gains. We also do not make the assumption that the information providers are interested in the product that will later be generated using their information.

Yet another area in which information elicitation is implemented is polling. The information market [20, 21] approach has been suggested as a way to generate more reliable predictions than can be achieved with 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).

A somewhat different sub-field of information elicitation deals with eliciting information from humans [22, 23]. The challenges here are to model as accurately as possible the desires of people (as utility functions, for example) and to overcome some of the irrationality that affects human behavior and reporting. The reports that these schemes often rely upon can be noisy and even conflicting.

An economic analysis of information as a trade commodity within large markets has also been performed. Broker Agents that buy information, filter it, and then sell the results have been examined in [24]. [25] explores the effects of

[4] This has led to controversy between Bayesians and Frequentists.

bundling information goods together.

Another example of the treatment of information elicitation appears in [26], where a manager in some firm attempts to obtain information from an employee in a setting where obtaining this information is costly. The manager attempts to create the incentive for truthful revelation by comparing it with his own information. The main result in [26] shows that if the worker has some signal on the information of the manager, it may become a "yes-man" that attempts to correlate the information it reports with the information of the manager, instead of reporting truthfully.

The automatic design of general mechanisms has been researched as well. [27, 28] 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. Here we propose to do similar things with information elicitation mechanisms.

In Section 5 we present mechanisms that use partial revelation of information. This agenda has also been pursued within preference elicitation settings [29, 30]. There, mechanisms are designed to approximately implement truth-telling in dominant strategies using constrained optimization techniques that are similar to our own.

2.2 Strictly Proper Scoring Rules

Scoring rules [10] 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: $S : \mathcal{P} \times \Omega \to \mathbb{R}$.

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, when the actual event ω is drawn from the probability distribution p (which we denote by $\omega \sim p$) the expected score of the predictor is higher if it reports p rather than any other distribution q:

$$E_{\omega \sim p}[S(p, \omega)] \geq E_{\omega \sim p}[S(q, \omega)] \tag{1}$$

In the above, equality is achieved iff $p = q$. [12] show a necessary and sufficient condition for a scoring rule to be strictly proper (see a generalized version in [11]), which allows easy generation of various proper scoring rules by selecting a bounded convex function over $\mathcal{P}$. Each such function generates a new scoring rule.

Several commonly known scoring rules are:

- The spherical scoring rule:

$$S(p, \omega) = \frac{p_\omega}{\sqrt{\sum\limits_{\omega' \in \Omega} {p_{\omega'}}^2}} \tag{2}$$

- The logarithmic scoring rule:

$$S(p,\omega) = log(p_\omega) \tag{3}$$

- And the quadratic scoring rule:

$$S(p,\omega) = 2p_\omega - \sum_{\omega' \in \Omega} p_{\omega'}^2. \tag{4}$$

An interesting use of scoring rules within the context of a multiagent reputation system was suggested by [15], who have modeled the bad behavior of service providers by a random variable that, with some fixed probability p, determines whether they will be honest or dishonest in their next transaction. A series of agents interact with this service provider; each is required to give feedback, which is interpreted as giving some refined prediction for the value of p. An agent involved in giving feedback is then rewarded with a scoring rule according to how well it predicted the feedback signal of the next agent that interacts with the service provider. This mechanism makes true revelation of the experience with the service provider a Nash equilibrium. Unavoidably, the mechanism also has other Nash equilibria that may attract agents. This may be corrected by relying on some reliable feedback from other sources as well [31].

2.3 Stochastic Programming

Stochastic Programming [32] 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 work, we use the conservative formulation of [33] which requires the assignment of variables to satisfy the constraints of the program for *every* possible program instance. For example, if we are given a program of the form:

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

where A is considered to be from an allowed set of parameters $\mathcal{A}$, we shall require a solution x to the mathematical program to be feasible for all possible $A \in \mathcal{A}$.

This type of linear stochastic program has a convex solution space (it is the intersection of a convex space for every possible $A \in \mathcal{A}$). General convex optimization algorithms require a description of the solution space, e.g., via a separation oracle. A separation oracle is simply a program that is able to tell if a certain point x is in the solution space, and if it is not, can provide a linear separator between the set of allowed solutions and x. If an efficient separation oracle exists, the convex optimization problem can be solved efficiently as well. An efficient oracle can be constructed for the stochastic optimization problem above using a linear program solver (see [33] for more details), and thus it is efficiently solvable.

We shall make use of this formulation later in Section 4. Each instance will correspond to a different variation in the beliefs held by the participating agents.

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. We look at a different problem: we assume each agent (including the principal, i.e., the one trying to elicit the information) 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 differing cases. Bartered knowledge is very often presented in a discrete format.[5] Finally, aggregating information from several agents is also much clearer and simpler to do with discrete variables.

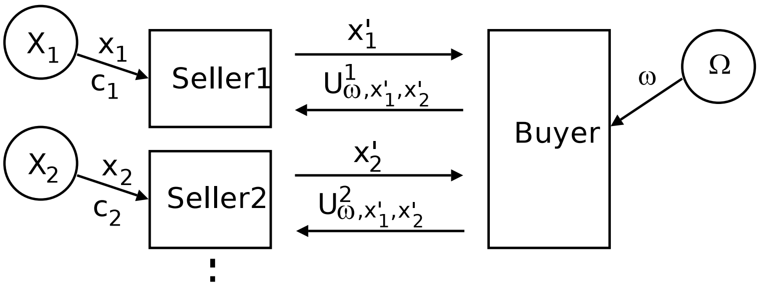


Figure 1: The Information Transaction

We assume the buyer wishes to purchase information about the value of a discrete random variable X_i from each seller i, and that the seller can learn the value of that variable at a cost c_i. To verify the quality of the information

[5] For example, in the original example from Section 1.1 above, Bob could be interested in knowing the condition of his car but would not really care for a continuous range of values. The required information in this case might be given just to make a discrete choice of whether or not to purchase more insurance. Continuous data can sometimes be made discrete according to the various actions it implies: if the car is in a condition that is worse than some threshold, Bob would purchase insurance, otherwise he would not. This defines the discrete information in which he is interested.

it purchases, the buyer has access to a random variable Ω. $\Omega, X_1, \ldots, X_n$ are presumably not independent variables, and knowledge about the value of one of them gives some information regarding the value of the others. Using the variable Ω, the buyer can get some idea if the information sold to him was correct. Without Ω, it would sometimes be impossible to create the necessary incentives for truthfulness on the part of the sellers. The variable may be redundant in the case of multiple sellers where information from several sources can be compared for validation.

We shall denote the probability distribution for $\Omega, X_1, \ldots, X_n$ by $p_{\omega,x_1,\ldots,x_n} = Pr(\Omega = \omega, X_1 = x_1, \ldots, X_n = x_n)$. The values the different variables can take, as well as the probability distribution $p_{\omega,x_1,\ldots,x_n}$, and the costs c_i are assumed to be common knowledge. We also assume that agents seek to maximize their expected gains and that they are risk-neutral.

In our running example, the variable X that Bob wishes to purchase is the mechanical state of the car, the verification variable Ω is the indicator that denotes if Bob's car breaks down within six months of being evaluated, and the probability distribution $P_{\omega,x}$ links these two events. A car in bad condition has a higher chance of breaking down. The cost c represents the amount of work Alice has to invest to examine Bob's car.

The buyer can now design a payment scheme that will determine the payment it must give to the sellers, based on the information the sellers gave and on the value of the verification variable Ω. We shall denote the payment to agent i by $u^i_{\omega,x_1,\ldots,x_n}$.

In our example, this payment scheme captures the different payments upon which Bob and Alice agree. If Alice says the car is fine, Bob will pay her more if the car does not break down (and may even claim money if it does), while a different set of payments will apply if Alice says the car is in poor condition.

It is important to stress that the variable Ω must be hidden from the sellers at the time the transaction is carried out. If the sellers possess too much information regarding Ω (in addition to what is implied through the information they sell), they may choose to report a value that best fits the buyer's signal instead of the real value they learned.

In this paper we primarily examine the restricted case of a single seller. The approach we take can be easily extended to multiple sellers.[6] 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 tell the true value that they found. These three requirements are defined more precisely below.

[6] With multiple sellers the mechanism designer has to make a decision regarding the exact solution concept the mechanism will use. A wide range is available, for example, a dominant strategy implementation, an iterated dominance implementation, or a Nash equilibrium implementation. Each choice produces different constraints, but all are similar in spirit to the formulation we present for the single agent. Even more complex mechanisms can be designed to resist various forms of collusion among multiple sellers.

3.1 The Requirements from the Mechanism in the Single Agent Case

In the case of one participating agent with a single variable, we need to satisfy three types of constraints in order to have a proper 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'.

$$\forall x, x' \quad s.t. \quad x \neq x' \quad \sum_{\omega} p_{\omega,x} \cdot (u_{\omega,x} - u_{\omega,x'}) > 0 \tag{5}$$

   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.

   In our example this is the requirement that Alice tells Bob the truth about the state of the car in the event that she knows it.

2. **Individual Rationality.** An agent must have a positive expected utility from participating in the game:

$$\sum_{\omega,x} p_{\omega,x} \cdot u_{\omega,x} > c \tag{6}$$

   This assures us that Alice will want to do business with Bob. If she does not stand to gain from the transaction (even in expectation), she will prefer not to deal with Bob at all.

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:

$$\forall x' \quad \sum_{\omega,x} p_{\omega,x} \cdot u_{\omega,x} - c > \sum_{\omega,x} p_{\omega,x} \cdot u_{\omega,x'} \tag{7}$$

   This constraint will ensure that Alice will be better off making an informed judgment regarding Bob's car rather than just guessing its mechanical condition.

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: $\sum_{\omega,x} p_{\omega,x} \cdot u_{\omega,x}$.

Let us demonstrate a proper mechanism using a numeric example:

***Example* 1.** *Let us assume that Bob's car can be in only one of two states: good working condition or poor working condition,* $X = \{good, poor\}$*, and that he wishes to find out which state it is in. He then turns to Alice who can invest* *an effort that is equal to \$10 to find this out. Bobs wants Alice to tell the truth so he conditions payments on the event in which the car breaks down (or does not) in the next 6 months:* $\Omega = \{break\ down, ok\}$*.*

The probability distribution, which is assumed to be common knowledge, is:

Car Condition	*Break Down*	*Probability*
good	*no*	*0.4*
good	*yes*	*0.1*
poor	*no*	*0.3*
poor	*yes*	*0.2*

Note that this distribution implies that the car has an equal probability of being in good condition or of being in poor condition, and that without knowing the condition of the car, it has a 0.15 probability of breaking down. Now we assume that Alice and Bob decided on the following payment scheme:

Reported Condition	*Break Down*	*Payment*
good	*no*	*\$15*
good	*yes*	*\$0*
poor	*no*	*\$0*
poor	*yes*	*\$25*

Now, let us check that each one of the constraints is satisfied:

1. **Truth Telling.** *Let us assume that Alice knows the car is in good shape. The car therefore has a 20% chance of breaking down. If Alice reveals the truth to Bob, she will get paid \$15 with probability 0.8 (the case where the car does not break down—an expected value of \$12). Otherwise, she may lie and report the car is in poor shape, and get paid only if it breaks down; that means getting \$25 with probability 0.2, which is worse (\$5 in expectation). So in this case, Alice would be better off telling the truth.*

   The reader may verify that Alice will also want to tell the truth if she knows the car is in poor condition (a 40% chance of breaking down). In this case, telling the truth gives her \$25 with probability 0.4 (or \$10 in expectation) and lying will give her \$15 with probability 0.6 (\$9 in expectation), so she will tell the truth.

2. **Individual Rationality.** *If Alice checks the car and tells the truth, she is expected to get \$9 if the car is in poor shape and \$12 if it is in good shape. Since both events have equal probability, she stands to gain \$10.5 in expectation, which is higher than the effort she needs to invest in checking the car.*

3. **Investment.** *If Alice decides not to invest effort, then she can estimate the car's probability of breakdown at 0.15. If she tells Bob the car is in good shape, she stands to make \$15 with 85% probability, which is more than her expected value if she tells the truth (also because she has to invest \$10 worth of effort in that case). She may therefore decide not to check the condition of the car, and just tell Bob that it is fine.*

Due to the last constraint being violated, we see that this payment scheme will probably not *elicit the truth from Alice. However, as we shall later prove, a proper mechanism does exist.*

3.2 A Geometric Interpretation for the Truth-Telling Constraints

In Section 3.3 we shall see that if the truth-telling constraints are satisfiable, the payments can be adjusted easily to satisfy the rest of the constraints as well. We are therefore interested in better understanding these constraints.

A close look at the truth-telling constraints for some x and x',

$$\sum_{\omega} p_{\omega,x} \cdot (u_{\omega,x} - u_{\omega,x'}) > 0 \tag{8}$$

reveals that they seem similar to vector multiplication. In fact, if we define vectors

$$\vec{p}_x \triangleq (p_{\omega_1,x} \dots p_{\omega_k,x}) \tag{9}$$

$$\vec{u}_x \triangleq (u_{\omega_1,x} \dots u_{\omega_k,x}) \tag{10}$$

we can write the truth-telling constraints in the following form:

$$\forall x \neq x' \quad \vec{p}_x \cdot (\vec{u}_x - \vec{u}_{x'}) > 0. \tag{11}$$

Using a slightly different notation we can define:

$$\forall x \neq x' \quad \vec{v}_{x,x'} \triangleq \vec{u}_x - \vec{u}_{x'}, \tag{12}$$

and write the constraint as:

$$\forall x \neq x' \quad \vec{p}_x \cdot \vec{v}_{x,x'} > 0. \tag{13}$$

This representation has a geometric interpretation: the vector $\vec{p}_x$ is required to be on the positive side of the unbiased hyperplane perpendicular to the vector $\vec{v}_{x,x'}$.

It is important to notice that the vectors $\vec{v}_{x,x'}$ are not independent of each other, but have the following relationships:

$$\vec{v}_{x,x'} = -\vec{v}_{x',x} \tag{14}$$

$$\vec{v}_{x,x''} = \vec{v}_{x,x'} + \vec{v}_{x',x''} \tag{15}$$

We therefore have a matching requirement to 13 that places the vector $\vec{p}_{x'}$ on the negative side of the hyperplane $\vec{v}_{x,x'}$:

$$\forall x \neq x' \quad \vec{p}_{x'} \cdot \vec{v}_{x,x'} < 0 \tag{16}$$

A proper assignment of payments is required to give a linear separation between the vectors $\vec{p}_x$ and $\vec{p}_{x'}$ using the hyperplane defined by $\vec{v}_{x,x'}$ (see Figure 2). This requirement for linear separation is the basis for many of our results.

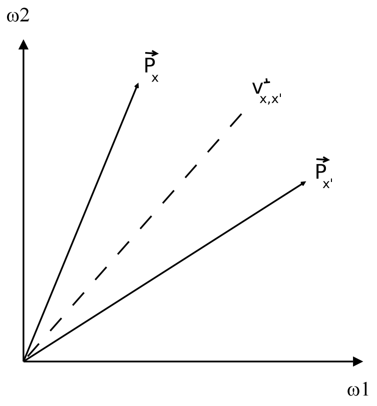


Figure 2: A Linear Separation of vectors $\vec{p}_x$ and $\vec{p}_{x'}$

3.3 Existence and Properties of Solutions for a Single Agent

There are naturally cases when it is impossible to satisfy the constraints. For example, if Bob's car is just as likely to break down no matter what condition it is in, we cannot expect that Bob will be able to create the incentive for Alice to tell the truth about her examination of the car. Bob will not be able to tell (even probabilistically) if she told the truth or not, and no mechanism will help. When can we be sure that a mechanism will exist?

The following proposition gives a sufficient condition for the existence of a mechanism 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 (according to Equation 5 and Equation 16):

$$0 < \sum_{\omega} p_{\omega,x} \cdot (u_{\omega,x} - u_{\omega,x'}) < 0 \tag{17}$$

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'. If the vectors are dependent, they cannot be linearly separable as required by the constraints. We shall later see that a high similarity between these vectors which makes them harder to separate, while still allowing for a working mechanism, actually limits its robustness.

Next, we show that if the condition described in Proposition 1 does not hold, we can always construct a proper payment scheme. Moreover, once we have some working payment scheme, we can easily turn it into an optimal one for the principal 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.*

Intuitively, this means that if the state of the car influences the chances of it breaking down even to a very small degree, then Bob can find a payment scheme that will properly motivate Alice to tell the truth. The proof idea is that once the truth-telling constraints are satisfied (using a regular scoring rule), the other constraints can also be satisfied by scaling the payments, and adding a constant to them.

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

$$u_{\omega,x} = \alpha \cdot S(Pr(\omega|x), \omega) + \beta_\omega \tag{18}$$

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 (Equation 1) of the incentive for truth-telling regardless of the 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

$$\alpha > \max_{x'}\left[\frac{c}{\sum\limits_{\omega,x} p_{\omega,x}(S(Pr(\omega|x), \omega) - S(Pr(\omega|x'), \omega))}\right] \tag{19}$$

satisfies that constraint for every x'. This is also shown in [15].

Finally, we can use the β_ω values to satisfy the remaining individual rationality constraint tightly by shifting the payments until their average is just above c:

$$\beta_\omega = \beta > c - \alpha \sum_{\omega,x} p_{\omega,x} \cdot S(Pr(\omega|x), \omega) \tag{20}$$

□

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. Notice that we are able to achieve the optimal cost of c by allowing negative payments to the seller as well (penalties). If we allow only positive payments, the cost will be higher.

3.3.1 Bad Verifiers

We have seen that if the information being sold has no bearing on the distributions of the probabilistic verifier Ω, no payment scheme can possibly create the incentives we require. But what if Ω provides only a slight indication of the correctness of the information?

In our example, this would be a scenario where Bob's car is only slightly more likely to break down if it is in poor condition. How does that affect the mechanism that is to be constructed? We will show below that Bob will need a large difference between payments made to Alice, that will in fact increase the risk involved for both of them.

We are given a hint of this by the construction of the mechanism above. In order to satisfy the investment constraints, we needed to scale the payments and thus increase the risk level of the mechanism. This becomes more severe if the verifier variable is poorly correlated with the purchased information. It seems that when Ω is a weak verifier, the difference between payments must increase. This increase in the risk of payments causes the *value of information* for the seller to increase as well—up to a level in which it is worthwhile to make the effort and obtain the true information. We demonstrate this fact here for the case of $|X| = 2$.

***Example* 2** (Bad verifiers result in high risk mechanisms)**.** *Let us assume that Ω is indeed a poor verifier. The probabilities $Pr(\Omega \,|\, x1)$ and $Pr(\Omega \,|\, x2)$ must be very similar. Let us denote:*

$$\vec{p}_{x1} = \vec{q} + \vec{\epsilon} \quad ; \quad \vec{p}_{x2} = \vec{q} - \vec{\epsilon} \tag{21}$$

where $\vec{\epsilon}$ is a very small vector. The investment constraints for this case are therefore:

$$(\vec{q} + \vec{\epsilon}) \cdot \vec{u}_{x1} + (\vec{q} - \vec{\epsilon}) \cdot \vec{u}_{x2} > (\vec{q} + \vec{\epsilon}) \cdot \vec{u}_{x1} + (\vec{q} - \vec{\epsilon}) \cdot \vec{u}_{x1} + c \tag{22}$$

$$(\vec{q} + \vec{\epsilon}) \cdot \vec{u}_{x1} + (\vec{q} - \vec{\epsilon}) \cdot \vec{u}_{x2} > (\vec{q} + \vec{\epsilon}) \cdot \vec{u}_{x2} + (\vec{q} - \vec{\epsilon}) \cdot \vec{u}_{x2} + c \tag{23}$$

Combining them gives us:

$$2\vec{\epsilon} \cdot \vec{u}_{x1} > 2\vec{\epsilon} \cdot \vec{u}_{x2} + 2c \tag{24}$$

which simplifies to:

$$||\vec{u}_{x1} - \vec{u}_{x2}|| \cdot ||\vec{\epsilon}|| \geq (\vec{u}_{x1} - \vec{u}_{x2}) \cdot \vec{\epsilon} > c \tag{25}$$

From this last inequality we see that as the norm of $\vec{\epsilon}$ goes to 0, the difference between the payment vectors $(\vec{u}_{x1} - \vec{u}_{x2})$ goes to infinity—which indicates a high level of variation in payments dictated by the mechanism. If we add the restriction of paying only positive payments, this implies that the expected cost of the mechanism goes to infinity as well.

4 Belief-Robust Mechanisms

In the previous section, we saw that it is easy to design information elicitation mechanisms in the single agent case. However, we assumed that the mechanism designer has precise knowledge about the probability distribution $p_{\omega,x}$, and that the seller of information is using the exact same distribution while it is contemplating which action to take. This is generally an assumption that is unlikely to hold.

In our running example, the mechanic Alice may have more expertise and may assign a probability to the event in which Bob's car breaks down that is different than Bob's uninformed assessment. Can Bob still assign payments that will properly motivate Alice? What will be the cost of this missing knowledge?

In many real-world scenarios, probabilities are often assessed through modeling or sampling (Alice can know the chance of a car breaking down more accurately because she has encountered more cars, or has a better understanding of why and when they break down), and two agents may have two different notions of the probabilities of certain events. This could have serious effects on the reliability of mechanisms designed for real systems.

We shall therefore try to 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. This assumption is reasonable, for example, in cases where distributions are learned by sampling and past experience. If some event has probability p of occurring, two agents sampling independently will not disagree greatly about that probability.

We denote the beliefs of the mechanism designer by $\hat{p}$ and the beliefs of a participating agent by $p = \hat{p} + \epsilon$, where ϵ is small according to some norm. We have opted for the L_∞ norm in this work, because it is easily described using linear constraints (it simply takes the maximum over all coordinates). Other norms may also be used, and will yield convex optimization problems that are not linear.

Next, we define the notion of belief robustness of the mechanism and through it examine the design of mechanisms that are still expected to work even if there is some difference between the beliefs of agents. We argue that not all payment schemes are equal—some may be more robust to changes in beliefs than others and should therefore be the preferred choice for use in real-world domains.

4.1 The Robustness Level of a Payment Scheme

Figure 3 presents a case in which the probabilities the seller believes in are not exactly known and may be within a certain region around what the buyer believes. The two payment schemes portrayed, v' and v'', are not the same. The scheme denoted by v' ensures that the probability vectors will be linearly separated (as is required by Equation 13), while v'' may fail to do so in some cases. We shall therefore want to think of v' as a more robust payment scheme than v''. Whenever the perturbation is severe and v'' does not linearly separate the vectors, the buyer of information will be lied to by the seller that has different beliefs and may conclude that lying is beneficial.

***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*

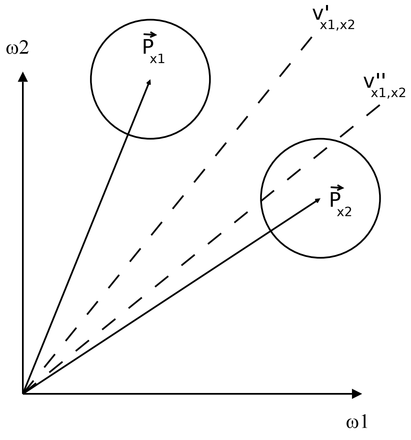


Figure 3: An elicitation problem with uncertain probabilities, and two payment schemes with different robustness levels.

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 that the mechanism work for every possible difference in beliefs. Another possible approach is to use an explicit probability distribution over possible continuous beliefs of the agents involved and require that the mechanism work well in a large-enough portion of the cases.[7]

Intuitively, it is appealing to attempt to find a maximal margin separation between the vectors, and use that to construct the payment schemes in a robust way. We show in Section 4.2.1 that this is indeed related to our definition of robustness, and discuss why such an approach will not work directly.

4.1.1 Determining the Robustness Level of a Mechanism

Given an offer for a payment mechanism from Alice, Bob can check to see if it will create the incentives for Alice (according to his own beliefs). What is the level of change in beliefs that will still keep the mechanism proper?

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. We are given the values of the payments and use them as parameters in the program to find a minimal perturbation of the probabilities

[7] This alternative formulation can also be handled using tools similar to those we use here (i.e., stochastic programming). Later, in Section 5, we examine a situation where there is a discrete set of possible beliefs held by the seller.

that will violate some constraint. For example, we can write the following program to find the worst case for one of the truth-telling constraints:

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

In the program above, only ϵ and $\epsilon_{\omega,x}$ are variables. The linear programs for other constraints are easily built by substituting, for the first constraint above, the negation of one of the constraints in the original design problem:

$$
\begin{array}{rl}
\min & \epsilon \quad \text{s.t.} \\
 & \{\text{place the negation of one of the constraints here}\} \\
\forall x, \omega & \hat{p}_{\omega,x} + \epsilon_{\omega,x} \geq 0 \\
 & \sum\limits_{\omega,x} \epsilon_{\omega,x} = 0 \\
\forall x, \omega & -\epsilon \leq \epsilon_{\omega,x} \leq \epsilon
\end{array}
$$

Once we have solved similar linear programs for all the constraints in the original design problem (a total of $|X|^2 + 1$ linear programs to solve), 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. We solve several programs here instead of just one large program because when formulating the problem this way it can be solved using linear optimizers, which are often simpler than general convex optimizers.

4.1.2 Finding a Mechanism With a Given Robustness Level

Now that Bob realizes that his original mechanism is not robust enough, he may try to find one that would be (if such a mechanism exists). It is possible to search for a payment scheme with a given robustness level ϵ using the following stochastic program:

	min	$\sum_{\omega,x} \hat{p}_{\omega,x} \cdot u_{\omega,x}$	Target function
s.t.	$\forall x \neq x'$	$\sum_{\omega} p_{\omega,x}(u_{\omega,x} - u_{\omega,x'}) > 0$	
		$\sum_{\omega,x} p_{\omega,x} \cdot u_{\omega,x} > c$	Constraints
	$\forall x'$	$\sum_{\omega,x} p_{\omega,x}(u_{\omega,x} - u_{\omega,x'}) > c$	
where:	$\forall x, \omega$	$p_{\omega,x} = \hat{p}_{\omega,x} + \epsilon_{\omega,x}$	
		$p_{\omega,x} \geq 0 \quad ; \quad \sum_{\omega,x} p_{\omega,x} = 1$	Parameter Range
		$-\epsilon \leq \epsilon_{\omega,x} \leq \epsilon$	

In this program, the variables are the payments $u_{\omega,x}$, while the probabilities $p_{\omega,x}$ are parameters that are unknown but are within some limited distance from $\hat{p}_{\omega,x}$. The program considers all distributions p that are close to $\hat{p}$ up to ϵ, according to the L_∞ norm. As we have mentioned before (in Section 2.3), this problem is convex.

In fact, we have already seen how to build a separation oracle for it—given a payment scheme $u_{\omega,x}$ we can check its robustness as shown in Section 4.1.1. This check will tell us if our payment scheme is within the allowed convex area. If it is not, it will provide us with a perturbation $\epsilon_{\omega,x}$ for which the solution fails. This gives us a linear condition that all solutions are required to uphold, but the given scheme does not (and is exactly what a separation oracle is required to provide). This procedure is called constraint generation and is often used in optimization problems. More details can be found in [33].

If a norm that is different than the L_∞ norm is used, the parameters section in the stochastic program would be different: the perturbation $\vec{\epsilon}$ would still be required to reside within a ball of some radius (only it is a ball according to some other norm) which is also a convex shape. In this case, if we want to check if some perturbation violates any of the constraints, we will have to use a convex program solver to search this ball for such a perturbation. The only difference is that this convex program is no longer a linear program as we were assured when we used L_∞, but it is still efficiently solvable.

4.1.3 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.

4.2 The Robustness Level of an Elicitation Problem

Bob may want to get the truth from Alice more than he cares about saving money. In cases like this, he may want the most robust payment scheme he can find. This in a sense is the best mechanism he can compose with his limited information about the probabilities of events. Is there a limit to the robustness that can be obtained? How can it be computed?

We 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:*

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

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. We achieve this in a manner similar to Equations 19 and 20. We simply scale the solution to give robustness for the investment constraint, and shift it to add robustness to the incentive compatibility constraint. Since the solution is ϵ-robust for the truth-telling constraints we have:

$$\forall \vec{\epsilon} \quad s.t. \quad \|\vec{\epsilon}\| < \epsilon \quad \forall x \neq x' \quad \sum_{\omega} p_{\omega,x}(u_{\omega,x} - u_{\omega,x'}) > 0. \tag{26}$$

If we sum over x we get:

$$\sum_{\omega,x} p_{\omega,x}(u_{\omega,x} - u_{\omega,x'}) > 0 \tag{27}$$

which implies that there exists a number $\delta_{x',\vec{\epsilon}}$ such that:

$$\sum_{\omega,x} p_{\omega,x}(u_{\omega,x} - u_{\omega,x'}) > \delta_{x',\vec{\epsilon}} > 0. \tag{28}$$

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 a new payment scheme $\tilde{u}$ for which:

$$\forall \vec{\epsilon} \quad \forall x' \quad \sum_{\omega,x} p_{\omega,x}(\tilde{u}_{\omega,x} - \tilde{u}_{\omega,x'}) > c \tag{29}$$

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 regard to the previous constraints. We can simply add a constant β to every payment:

$$\beta > c - \min_{\omega,x}[\tilde{u}_{\omega,x}]. \tag{30}$$

We will thus get a solution u^* that satisfies

$$\forall \omega, x \quad u^*_{\omega,x} > c \tag{31}$$

and therefore satisfies

$$\sum_{\omega,x} p_{\omega,x} \cdot u^*_{\omega,x} > \sum_{\omega,x} p_{\omega,x} \cdot c = c \tag{32}$$

for all possible belief changes, meaning that u^* is ϵ-robust. □

4.2.1 A Bound for Problem-Robustness

A simple bound for robustness of the problem can be derived from the requirement of eliciting the truth between just two possible statements the information seller may provide. This shows the relation between maximal margin separators and our notion of robustness. Proposition 1 for non-robust mechanisms can be viewed as a specific case of the following proposition (when applied to 0-robust mechanisms):

***Proposition* 4.** *The robustness level ϵ^* of a problem $\hat{p}$ can be bounded by the distance between any vector $\hat{p}_x$ and the optimal hyperplane that separates it from any other vector $\hat{p}_{x'}$. By selecting the pair of vectors that minimizes this bound we get:*

$$\epsilon^* \leq \min_{x,x'} ||\hat{p}_x - (\hat{p}_x^{tr} \cdot \vec{\varphi}_{x,x'}) \cdot \vec{\varphi}_{x,x'}||_\infty \tag{33}$$

$$\vec{\varphi}_{x,x'} = \frac{\hat{p}_x + \hat{p}_{x'}}{||\hat{p}_x + \hat{p}_{x'}||_2} \tag{34}$$

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

Here $\vec{\varphi}_{x,x'}$ is a normalized vector that passes within equal distance of $\hat{p}_x$ and $\hat{p}_{x'}$ (see Figure 4). ϵ^* is then limited by that equal distance, which we compute by subtracting from $\hat{p}_x$ its projection in the direction $\vec{\varphi}_{x,x'}$.

Proof. If there exist x, x' that are within a distance of ϵ to the hyperplane, then these two 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. □

It appears from the proposition above that the most robust mechanism could be found easily by finding optimal separating hyperplanes between the probability vectors $\vec{p}_x$, but in fact this alone will not do. The payment vectors that define the separators have additional constraints (see Equations 14 and 15) that relate the various separators to one another. In addition, one must consider robustness with regard to other constraints (not just the truth-telling constraints). With these additional factors, the most robust payments may not match the optimal separators at all.

However, 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—then the problem robustness is determined exactly by the closest pair of vectors.

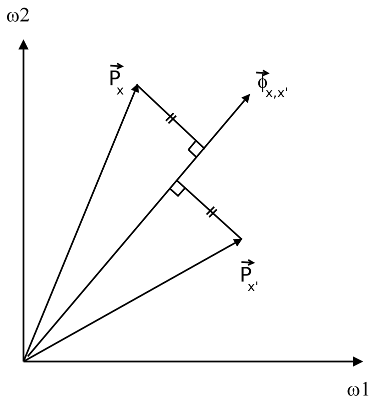


Figure 4: A Bound for the Robustness Level

5 Partial Revelation Mechanisms

In this section we shall explore, from a different angle, the problem of a common prior between agents. We shall modify our model of the information transaction and give the seller of information an extra random variable S that it can access. The value of S will not be divulged to the buyer, but may influence the decisions of the seller. We will however assume that the buyer is aware of the existence of this extra information and its possible values. As a result of this extra information, the buyer is placed at a disadvantage. It knows even less about the state of the world than the seller. This condition holds even after the transaction is concluded.

For example, in the scenario we presented in Section 1.1, Charlie who is Alice's boss may design the payment scheme so that Alice tells all of her customers the truth. He knows that Bob's car may have one of two different motors installed in it. One motor is of higher quality and has less chance of breaking down, and the other is of lower quality. Alice, who is a trained mechanic, can find out which type of motor Bob has (which she can do at a mere glance without any effort), which will influence the probabilities she associates with a malfunction. Bob may not be interested in the exact make of his engine, only in the likelihood that the car will break down. Can Alice still be financially encouraged to tell the truth?

Figure 5 describes the new elicitation scenario. The seller still needs to pay a cost of c to access the random variable X and report its findings to the buyer, only now it can access (for free) the random variable S as well. The payment made by the buyer depends only on the information it has available—not on the value of S. Once again we assume that a probability distribution $p_{\omega,x,s}$ governs the three variables, and that it is common knowledge. Note however, that since the seller alone has access to S, it has a clearer and more precise knowledge of the distribution of X and Ω since it knows $Pr(\Omega = \omega, X = x \,|\, S = s)$.

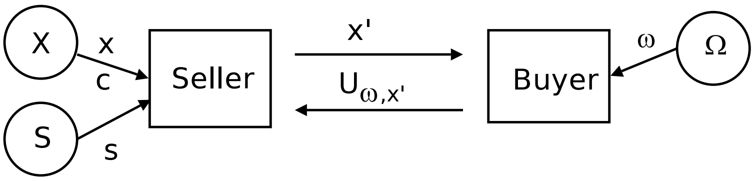


Figure 5: An Elicitation Scenario with a Secret Variable S

The three types of requirements from a mechanism are similar to those we have seen before. We shall now say that a mechanism is *proper for a secret s* if the following three conditions hold:

1. **Truth Telling.**

$$\forall x, x' \quad s.t. \quad x \neq x', \quad \sum_{\omega} p_{\omega,x,s} \cdot (u_{\omega,x} - u_{\omega,x'}) > 0 \tag{35}$$

2. **Individual Rationality.**

$$\sum_{\omega,x} p_{\omega,x,s} \cdot u_{\omega,x} > c \cdot p_s \tag{36}$$

3. **Investment.**

$$\forall x' \quad \sum_{\omega,x} p_{\omega,x,s} \cdot u_{\omega,x} - c \cdot p_s > \sum_{\omega,x} p_{\omega,x,s} \cdot u_{\omega,x'} \tag{37}$$

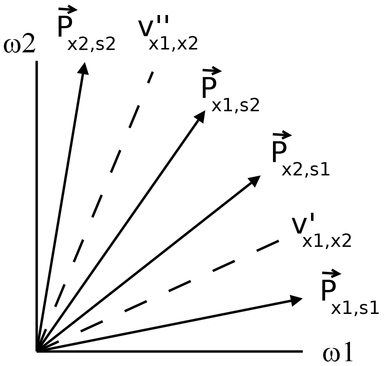
When designing partial revelation mechanisms, there are often probability distributions that do not allow us to construct an effective mechanism for *all* possible secrets the seller may hold. The example in Figure 6 demonstrates such a case.

It is impossible to find a separating hyperplane that will separate $\vec{p}_{x1,s1}$ from $\vec{p}_{x2,s1}$ and at the same time separate $\vec{p}_{x1,s2}$ from $\vec{p}_{x2,s2}$. The hyperplanes v' and v'' work only for a single secret each. Since the buyer is never told about the actual secret s, it has no way of creating the incentives for truthfulness in both cases.

We must therefore settle on building a mechanism that will work only part of the time. We will naturally aspire to have a good confidence level in our mechanism—to build a mechanism that will work with high probability. There are two possible alternatives we examine here:

1. A single-use, disposable mechanism—where we design the mechanism for only a single transaction. We then want the buyer's confidence in the received answer to be high:

$$\theta_1 = Pr_{s,x}(u \text{ is proper for state } (s, x)). \tag{38}$$



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

Figure 6: An Elicitation Scenario with Two Possible Results, and Two Possible Secrets

This means that we only require truth-telling in case of secret s and value x' that occur.

If we refer back to our example, this sort of mechanism will be appealing to Bob if he needs a single evaluation of the condition of his car. In this case, Alice may report several different findings. Some of them have very low probability (for example, the event of a crack in the motor may be extremely rare) and Bob may not mind if that specific piece of information is not elicited correctly (because it is such a remote occurrence).

2. A reusable mechanism—where we design the mechanism for multiple transactions. Here, we want the buyer to have high confidence that, once the secret s has been set, he will hear the truth for all possible cases of X:

$$\theta_2 = Pr_s(u \text{ is proper for secret } s). \tag{39}$$

Referring again to our example, this mechanism may have appeal if Bob wants several evaluations of his car performed, and wants the truth for as many of them as possible. Since the model of Bob's engine does not change between examinations of the car, he can take that into account when maximizing the probability of hearing the truth.

5.1 Complexity of Partial Revelation Mechanism Design

***Proposition* 5.** *Deciding if a reusable revelation mechanism with a confidence level over some threshold θ exists is NP-Complete. Furthermore, the problem of finding the mechanism with the maximal confidence level cannot be approximated within any constant.*

The design problem is in NP. This is because if we are given access to an oracle that tells us which secrets to try to satisfy and which to give up on, we can find a payment scheme that satisfies the right constraints in polynomial time. This is achieved by solving the linear program that consists of the constraints for all the included secrets.

We show that constructing a fully operational mechanism is NP-Complete by presenting a reduction from the Independent Set problem. The full reduction is presented in the appendix. The Independent Set problem, in addition to being NP-Complete, is also hard to approximate [34]. The reduction we give is a cost-preserving reduction and therefore demonstrates that our problem is just as hard to approximate as Independent Set.

The high complexity of designing proper mechanisms applies in the single-use, disposable case as well.

***Proposition* 6.** *Deciding if there exists a single-use elicitation mechanism with a confidence level over some threshold θ is also NP-Complete.*

Proof of this proposition relies on a reduction from the Hyperplane-Consistency problem. The full proof appears in the appendix.

5.2 Finding Partial Revelation Mechanisms

We now present two approaches to computing a partial revelation mechanism for a given problem $p_{\omega,x,s}$. As we have already seen, the problem of finding such a mechanism is NP-Complete, and unless P=NP, we cannot hope to locate the optimal mechanism in polynomial time in all cases. However, in some cases, the problem may be simpler than the worst possible case. The two approaches we present differ in the complexity of the algorithm. One algorithm will be better in cases where $|S|$ is small, while the other will be better in cases where $|\Omega| \cdot |X|$ is small.

The algorithms we present are for reusable mechanisms. Similar versions can be constructed for the single-use case.

5.2.1 Considering All Combinations of Secrets

The reductions we used in the proofs of Propositions 5 and 6 both relied on the difficulty of selecting the cases in which we wish the mechanism to work. This difficulty arises due to the discrete nature of the secrets. If we had an oracle that shows us which constraints to try to satisfy, we could easily construct a mechanism. Since we do not possess such an oracle, we can try every possible combination by brute force. This method relies on the discrete nature of the problem and the finite set of possible secrets:

In the algorithm above, there are $2^{|S|}$ ways to select secrets to satisfy. Each selection then requires $poly(|S||X||\Omega|)$ time to check for feasibility. This therefore gives a running time of $O(2^{|S|} \cdot poly(|S||X||\Omega|))$ which can still be efficient if the number of possible secrets is small.

Algorithm 1 [Reusable Mechanism Construction]:

1. For all $W \in 2^S$:
   (a) Locate a mechanism that satisfies all constraints for all secrets in W.
   (b) If such a mechanism exists, compute $\theta_W = \sum_{s \in W} p_s$.
2. Return a mechanism for secrets $\arg\max_W(\theta_W)$.

5.2.2 The Geometric Approach—Partitioning into Cells

The second approach we shall examine is based on a geometric interpretation of the problem. The linear constraint for the mechanism design problem partitions the space of payment vectors into cells. Each cell is a region of the space for which some set of constraints holds, while the rest are violated.

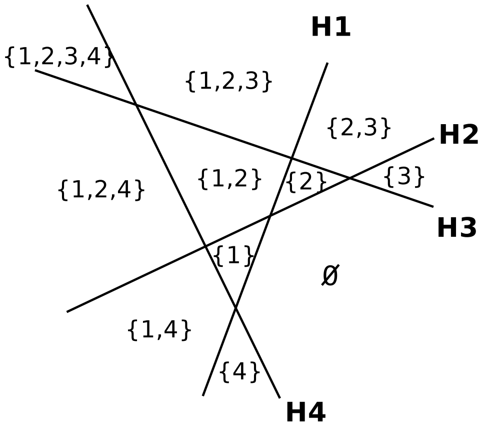


Figure 7: A collection of hyperplanes partitioning the plane into cells.

The mechanism design problem is in fact the problem of locating a non-empty cell that satisfies as many constraints as possible. This naturally leads to an algorithm that builds a list of cells and iterates over them to locate the cell assignment with the highest score.

In order to generate the list of cells L needed in the algorithm above, one can simply start from a list containing a single cell that contains the entire vector space and incrementally add hyperplanes. Each hyperplane that is added may partition a cell in the list into two cells, one on either side of the hyperplane, or may leave the cell intact. At every stage one only needs to iterate over the list of existing cells and check if they are split by the new hyperplane.

Algorithm 2 [Geometric]:

1. Construct a list L of cells created by all hyperplanes $\vec{p}_{x,s}$.
2. Select an assignment $\sigma : X \times X \rightarrow L$.
3. Try to solve the linear problem that consists of constraints placing $v_{x,x'}$ in the cell $\sigma(x, x')$, and satisfying
   $\vec{v}_{x,x'} = -\vec{v}_{x',x} \quad ; \quad \vec{v}_{x,x''} = \vec{v}_{x,x'} + \vec{v}_{x',x''}$
4. If a solution is found, compute:
   (a) $W_\sigma \in 2^S$, the list of secrets that assignment σ of vectors $v_{x,x'}$ satisfies.
   (b) $\theta_\sigma = \sum\limits_{s \in W_\sigma} p_s$.
5. Return the payment scheme found for $\arg\max\limits_{\sigma}(\theta_\sigma)$.

5.2.3 Complexity of the Algorithm

In order to analyze the running time of Algorithm 2, we need to obtain a bound on the number of cells created by the hyperplanes defined by $\vec{p}_{x,s}$. Such a bound is given in [35]. Given m hyperplanes in d-dimensional space, the number of cells is bounded by:

$$\Phi_d(m) = \sum_{i=0}^{d} \binom{m}{i} = O(m^d). \tag{40}$$

The bound is obtained using the VC-Dimension of the concept class implied by cell partitioning and Sauer's lemma [36].

This bound is especially interesting when d is small, since it implies that the number of cells is only polynomial in the number of hyperplanes m.

In our case, we have $|X||S|$ hyperplanes in an $|\Omega|$-dimensional space, which gives a bound of $|L| = O(|X||S|^{|\Omega|})$ cells. Generating the list of cells can be done in

$$O(|X||S|^{2|\Omega|} \cdot poly(|X||S||\Omega|))$$

time steps. The number of possible assignments $\sigma : X \times X \rightarrow L$ is

$$O(|L|^{|X|^2}) = O(|X||S|^{|X|^2|\Omega|})$$

and for each assignment we need to solve a linear program that requires $poly(|X||S||\Omega|))$ steps, which gives us a total running time of

$$O(|X||S|^{|X|^2|\Omega|} \cdot poly(|X||S||\Omega|))$$

time steps.

This algorithm is therefore better in cases where $|S|$ is large, but $|\Omega|$ and $|X|$ are small.

6 Elicitation of Confidence Ratings

In many cases, the expert that sells the information has some idea regarding the reliability of the information it is selling. For example, a reviewer reading a paper is often asked to rate his or her familiarity with the field, and his or her confidence in the submitted review. Confidence level is also quite important when considering what to do with information—if a reviewer who is not confident was selected, the paper could be sent for further review to someone else. In other cases, only a noisy signal can be received—Alice may check Bob's car and discover that the problem is either in the ignition system, or the fuel injection system, but be unsure as to the exact origin.

It is therefore important to be able to elicit the confidence rating or degree of certainty regarding a piece of information, and not just the information itself. Fortunately, this is often possible with various models for inexact information, as these often reduce to regular information elicitation problems. We briefly demonstrate two such models here.

6.1 An Error Model

One such model for confidence would be to assume that with some probability p_c, the information learned by the expert is correct and drawn from the distribution $p_{\omega,x}$, while with probability $1-p_c$ it is erroneous and has a value of X according to some other distribution:

$$q_{\omega,x} = q(\Omega = \omega, X = x). \tag{41}$$

X and Ω may be independent in the distribution $q_{\omega,x}$, but this does not have to be the case.

The seller can then be asked to divulge p_c, as well as the value of X that it got. Figure 8 depicts the information transaction in this case. In this model, we allow p_c to take continuous values between 0 and 1, and assume that the cost of acquiring the information is 0, as there is no way to create the incentives to learn the value of X in case the confidence rating is $p_c = 0$.[8]

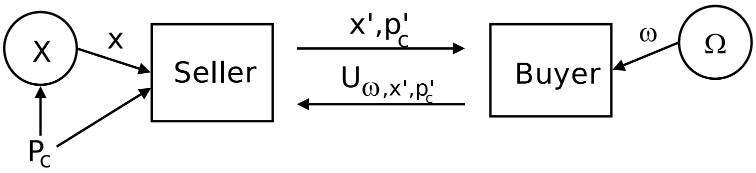


Figure 8: Elicitation of a Confidence Rating p_c

[8] If we are sure the confidence level is always higher than some positive number, then a mechanism can be designed to ensure investment of effort as well.

***Proposition* 7.** *The confidence ratings p_c and the true value of X can be elicited truthfully if there exists a truthful payment scheme for the elicitation of X with payments $\vec{u}_x$ such that for any $x, x' \in X$,*

$$\vec{q}_x \cdot (\vec{u}_x - \vec{u}_{x'}) \geq 0. \tag{42}$$

Proof. Let u be a payment scheme that truthfully elicits the value of X, as well as the condition from inequality 42. Now, given two possible values (x, p_c), (x', p'_c), if $x \neq x'$ we have:

$$\vec{q}_x \cdot (\vec{u}_x - \vec{u}_{x'}) \geq 0 \quad ; \quad \vec{p}_x \cdot (\vec{u}_x - \vec{u}_{x'}) > 0 \tag{43}$$

$$\vec{q}_{x'} \cdot (\vec{u}_x - \vec{u}_{x'}) \leq 0 \quad ; \quad \vec{p}_{x'} \cdot (\vec{u}_x - \vec{u}_{x'}) < 0 \tag{44}$$

which implies

$$(p_c \cdot \vec{p}_x + (1 - p_c) \cdot \vec{q}_x) \cdot (\vec{u}_x - \vec{u}_{x'}) \geq 0 \tag{45}$$

$$(p_{c'} \cdot \vec{p}_{x'} + (1 - p_{c'}) \cdot \vec{q}_{x'}) \cdot (\vec{u}_x - \vec{u}_{x'}) \leq 0 \tag{46}$$

where the inequalities above are strict whenever $p_c, p_{c'}$ are not 0. Now notice that $P(\Omega|x, p_c) = p_c \cdot \vec{p}_x + (1 - p_c) \cdot \vec{q}_x$ and Equations 45, 46 in fact show that the probabilities $P(\Omega|x, p_c)$ are different as long as $p_c, p_{c'} \neq 0$. We can then award the seller a payment according to some scoring rule:

$$u_{w,x,p_c} = S(P(\Omega = \omega|x, p_c), \omega).$$

Since the probabilities $P(\Omega|x, p_c)$ are different for every report, the scoring rule assures us of the incentive to tell the true value. □

***Remark* 1.** *The condition*

$$\vec{q}_x \cdot (\vec{u}_x - \vec{u}_{x'}) \geq 0$$

implies that the payment scheme u also truthfully elicits the value of X under the probability distribution q (as long as whenever the seller is indifferent between telling the truth and lying it will choose to tell the truth).

6.2 Inexact Knowledge

Another possible model in which the seller has unreliable information is one where instead of getting a value of $x \in X$ it receives a set $T \subset X$ from which the value of X will be chosen (and so it knows something more about the possible values of X, but not the exact value).

We then assume that learning the subset T implies that X will be chosen only from that subset according to the distribution $p_{\omega,x}$ that is reduced to T:

$$Pr(X = x, \Omega = \omega, x \in T) = \frac{p_{\omega,x}}{\sum_{\omega'} \sum_{x' \in T} p_{\omega',x'}} \tag{47}$$

In this case, we can ask the seller to provide the exact subset T that it learned. This problem once again is reduced to a truthful information elicitation problem, this time with T being the elicited variable. Each value of T implies a different distribution on the values of ω and can thus be elicited using mechanisms of the form we have shown in Section 3 for the distribution $Pr(\Omega|T)$.

Note, however, that the model we presented did not include information about the probability of a certain $T \subset X$ being selected, and that there is no way to discuss the elicitation of effort without such a model.[9]

7 Conclusions

We have introduced a model for discrete information transactions and have shown simple information elicitation mechanisms that can provide the sellers with the correct incentives to report honestly and even invest effort into obtaining the information they sell. We have shown that in most cases these simple mechanisms exist and can be designed optimally using scoring rules. We explored various properties of the solution, such as the cost of the mechanisms and the level of risk they entail when verification of information is difficult.

In order to tackle the problem of belief variations between the sellers and the designer of the mechanism, we introduced the concept of robust mechanisms. These mechanisms are guaranteed to work if the beliefs of agents are not too far apart. We have shown efficient algorithms for learning the robustness level of a given payment scheme, finding payment schemes with guaranteed level of robustness, and for finding the robustness level of a problem. The efficiency of their design, as well as their resilience, makes these mechanisms good candidates for application in real-world scenarios.

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 mainstream work done in stochastic programming.

To further explore information transactions, we examined a model in which the seller of information has access to extra information that is not sold. We have seen that partial revelation of information can make it impossible to build mechanisms that work all the time, and that building good mechanisms that work most of the time is a computationally difficult task. Here we have provided proofs of computational difficulty as well as two algorithms with different running times that may be suitable in different cases.

[9] A possible model is to assume T is selected according to a probability that is proportional to $\sum_{x \in T} p_x$, which then ensures that x is eventually selected according to the original distribution $p_{\omega,x}$.

8 Acknowledgments

The authors wish to thank the reviewers, whose comments provided guidance for our refining of the paper and its overall improvement. Preliminary material from this paper appeared at the Twenty-First National Conference on Artificial Intelligence (AAAI'06), in the papers "Robust Mechanisms for Information Elicitation" [37] and "Mechanisms for Partial Information Elicitation: The Truth, But Not the Whole Truth" [38]. This work was partially supported by Israel Science Foundation grant #898/05.

A Computational Hardness of Reusable Mechanism Design

Proposition: Deciding if a reusable revelation mechanism with a confidence level over some threshold θ exists is NP-Complete. Furthermore, the problem of finding the mechanism with the maximal confidence level cannot be approximated within any constant.

Proof. The proof relies on a reduction from the *Independent Set* problem. Given an undirected graph $G(V, E)$ and an integer $k \le |V|$ the *Independent Set* decision problem is defined as the problem of deciding whether there is a set of vertices $W \subset V$ so that $|W| \ge k$, and such that for every edge $e \in E$, e does not occur on more than one vertex in V.

The intuition behind the reduction is derived from Figure 6. The selection of vertices in the independent set will be designed to match a selection of secrets to satisfy in the design problem. In order to uphold the restriction that no two vertices sharing an edge can be chosen together, a construction similar to Figure 6 will be created in a dedicated two-dimensional subspace to assure that their matching secrets cannot be satisfied at the same time.

Let us now proceed with the proof. Given an instance of the Independent Set problem $(G(V, E), k)$ we shall construct a mechanism design problem $(\Omega, X, S, P, \theta)$ in the following manner:[10]

$$\Omega = \bigcup_{e \in E} \{\omega_{e1}, \omega_{e2}\} \quad ; \quad X = \bigcup_{e \in E} \{x_{e1}, x_{e2}\} \tag{48}$$

$$S = V \quad ; \quad \theta = \frac{k}{|V|} \tag{49}$$

We denote by $\vec{\delta}_i$ the vector that is 0 at all coordinates except for coordinate i, where it takes the value of 1, and by α a normalizing constant that equals $\alpha = \frac{1}{2|E||V|}$. P is then defined as follows:

[10] A small comment about notation: $e1$ and $e2$ above do not reference the two vertices of edge e, but just serve to denote two different values for that edge. We shall explicitly make it clear when we refer to vertices of the edge.

if $v \notin e$ then:

$$\vec{p}_{x_{e1},v} = \alpha \cdot \vec{\delta}_{\omega_{e1}} \quad ; \quad \vec{p}_{x_{e2},v} = \alpha \cdot \vec{\delta}_{\omega_{e2}} \tag{50}$$

otherwise $e = \{v1, v2\}$ and we set:

$$\vec{p}_{x_{e1},v1} = \alpha \cdot \vec{\delta}_{\omega_{e1}} \quad ; \quad \vec{p}_{x_{e2},v1} = \frac{\alpha}{2} \cdot (\vec{\delta}_{\omega_{e1}} + \vec{\delta}_{\omega_{e2}}) \tag{51}$$

$$\vec{p}_{x_{e1},v2} = \frac{\alpha}{2} \cdot (\vec{\delta}_{\omega_{e1}} + \vec{\delta}_{\omega_{e2}}) \quad ; \quad \vec{p}_{x_{e2},v2} = \alpha \cdot \vec{\delta}_{\omega_{e2}} \tag{52}$$

With the above construction all secrets have the same probability of occurring:

$$Pr(S = s) = \sum_{\omega,x} p_{\omega,x,s} = 2|E|\alpha = \frac{1}{|V|} \tag{53}$$

Below we show the two steps needed to complete the proof:

1. **If the graph G has an independent set of size k then there is a mechanism with a confidence level above the threshold θ.**

   Let us assume that G has an independent set $W \subset V$ of size k. We shall build a payment scheme that will give a proper mechanism for all the secrets matching the vertices in W. For an edge e that has one of its vertices in the independent set,[11] we shall define:

   $$\vec{u}_{x_{e1}} = \frac{\vec{p}_{x_{e1},v}}{||\vec{p}_{x_{e1},v}||} \quad ; \quad \vec{u}_{x_{e2}} = \frac{\vec{p}_{x_{e2},v}}{||\vec{p}_{x_{e2},v}||} \tag{54}$$

   where v is the vertex (from edge e) that was selected for the independent set. If on the other hand e did not have any vertex in the independent set, we simply set

   $$\vec{u}_{x_{e1}} = \vec{\delta}_{\omega_{e1}} \quad ; \quad \vec{u}_{x_{e2}} = \vec{\delta}_{\omega_{e2}} \tag{55}$$

   We will next demonstrate that this payment scheme does give a truthful mechanism at least for all the secrets in W. We must therefore show that

   $$\begin{aligned} &\forall v \in W \quad \forall x_{ek} \neq x_{e'l} \in X \\ &\qquad \vec{p}_{x_{ek},v} \cdot (\vec{u}_{x_{ek}} - \vec{u}_{x_{e'l}}) > 0 \end{aligned} \tag{56}$$

   Let us examine the following three cases:

   (a) Edge e does not occur on vertex v, and has no vertex in the independent set. Then the vector $\vec{p}_{x_{ek},v} = \alpha \cdot \vec{\delta}_{\omega_{ek}}$. Because e has no vertex in the independent set then $\vec{u}_{x_{ek}} = \vec{\delta}_{\omega_{ek}}$, meaning that it is a unit vector in the direction of $\vec{p}_{x_{ek},v}$. Since $\vec{u}_{x_{e'l}}$ is also a unit vector that is in the other direction, its inner product with $\vec{p}_{x_{ek},v}$ is smaller and the inequality holds.

[11] It cannot have both its vertices in the set—only one or none.

(b) Edge e does not occur on vertex v, but has another vertex in the independent set. In this case we still have $\vec{p}_{x_{ek},v} = \alpha \cdot \vec{\delta}_{\omega_{ek}}$ but now $\vec{u}_{x_{ek}}$ has two possible values, depending on which vertex of e is in the independent set. Either

   i. $\vec{u}_{x_{ek}} = \vec{\delta}_{\omega_{ek}}$

   ii. $\vec{u}_{x_{ek}} = \frac{1}{\sqrt{2}} \cdot (\vec{\delta}_{\omega_{ek}} + \vec{\delta}_{\omega_{e\bar{k}}})$

   In both these cases the inner product between $\vec{p}_{x_{ek},v}$ and $\vec{u}_{x_{ek}}$ is strictly positive. If e' is an edge that is different from e then $\vec{p}_{x_{ek},v} \cdot \vec{u}_{x_{e'l}}$ is zero and the inequality holds. Otherwise, $e' = e$ but $k \neq l$. In this case we observe that $(\vec{u}_{x_{ek}} - \vec{u}_{x_{e\bar{k}}}) \cdot \vec{\delta}_{\omega_{ek}} > 0$ (simply by looking at all the cases)—and this gives us the required inequality exactly.

(c) Edge e occurs on vertex v. Since we are only concerned with v's that are in the independent set, the vector $\vec{u}_{x_{ek}}$ is by our definition a unit vector in the direction of $\vec{p}_{x_{ek},v}$, while $\vec{u}_{x_{e'l}}$ is a unit vector in another direction. This means that the inner product between $\vec{p}_{x_{ek},v}$ and $\vec{u}_{x_{ek}}$ is greater and the inequality once again holds.

We have thus shown that we are able to have a working mechanism for every vertex $v \in W$ and thus have a mechanism that works well for k secrets. The confidence level of the mechanism designer in its mechanism is then at least $\sum\limits_{s \in W} Pr(S = s) = \frac{k}{|V|} = \theta$.

2. **If there is a good mechanism with confidence level above θ then there is an independent set of size k in the graph.**

   Since there is a confidence level of $\frac{k}{|V|}$, there must be at least k satisfied secrets in the mechanism. Each such secret matches a vertex in the original problem. It remains to show that the set W of vertices matching satisfied secrets is independent. Assuming the opposite leads to a contradiction. The secrets matching two vertices that are connected by an edge cannot be satisfied at the same time due to the way the problem was constructed. The probability vectors for each edge ($\vec{p}_{x_{e1},v1}$, $\vec{p}_{x_{e1},v2}$, $\vec{p}_{x_{e2},v1}$, $\vec{p}_{x_{e2},v2}$) were placed in a separate two-dimensional space, and were set similarly to the vectors in Figure 6—in a way that ensures that both pairs cannot be linearly separated at the same time.

   Let us show that the set W of vertices matching satisfied beliefs is independent. We first assume the opposite: that there are two vertices that we shall denote $v, v' \in W$ that reside on the same edge in G. By construction we therefore have two beliefs for vertices v, v' that were both satisfied. Meaning that

$$\forall x \neq x' \in X \quad \vec{p}_{x,v} \cdot (\vec{u}_x - \vec{u}_{x'}) > 0 \tag{57}$$

   and also that

$$\vec{p}_{x,v'} \cdot (\vec{u}_x - \vec{u}_{x'}) > 0 \tag{58}$$

More specifically, the above holds true for any specific values of x, x' that we choose, such as for x_{e1} and x_{e2}, where e is the edge that is shared by v, v'. Therefore, the following two statements must be true at the same time:

$$\vec{p}_{x_{e2},v} \cdot (\vec{u}_{x_{e2}} - \vec{u}_{x_{e1}}) > 0 \tag{59}$$

$$\vec{p}_{x_{e1},v'} \cdot (\vec{u}_{x_{e1}} - \vec{u}_{x_{e2}}) > 0 \tag{60}$$

Without loss of generality, we can assume at this point that v is the first vertex in edge e, and v' is the second vertex. Therefore, by construction we have:

$$\vec{p}_{x_{e2},v} = \vec{p}_{x_{e1},v'} = \frac{\alpha}{2} \cdot (\vec{\delta}_{\omega_{e1}} + \vec{\delta}_{\omega_{e2}}) \tag{61}$$

and by substituting this into the above, we reach a contradiction, since it cannot be the case that both of the following are true at the same time:

$$\vec{p}_{x_{e2},v} \cdot (\vec{u}_{x_{e2}} - \vec{u}_{x_{e1}}) > 0 \tag{62}$$

$$\vec{p}_{x_{e1},v'} \cdot (\vec{u}_{x_{e1}} - \vec{u}_{x_{e2}}) = -\vec{p}_{x_{e2},v} \cdot (\vec{u}_{x_{e2}} - \vec{u}_{x_{e1}}) > 0 \tag{63}$$

So our assumption that there can be two secrets v, v' that are satisfied at the same time but have counterpart vertices on the same edge is false, and the set W is indeed independent.

This completes the proof of the reduction. □

B Computational Hardness of Designing Single-Use Mechanisms

Proposition: Deciding if there exists a single-use elicitation mechanism with a confidence level over some threshold θ is NP-Complete.

Proof. We give a proof of this proposition using a series of reductions from the decision problem associated with Max-Hyperplane-Consistency. This problem is known to be NP-Complete [39].

An instance of the Hyperplane-Consistency problem is defined by a tuple $(\mathcal{P}, \mathcal{N}, k)$ where k is an integer and $\mathcal{P}, \mathcal{N}$ are sets of vectors in $\mathbb{R}^n$ with integer coordinates. The instance should be accepted if a biased linear separator $(\vec{w}, b)$ exists such that:

$$k \leq |\{\vec{x} \in \mathcal{P} \mid \vec{w} \cdot \vec{x} \geq b\}| + |\{\vec{x} \in \mathcal{N} \mid \vec{w} \cdot \vec{x} < b\}| \tag{64}$$

meaning that the separator $(\vec{w}, b)$ manages to place more than k points from $\mathcal{P}$ on its positive side and from $\mathcal{N}$ on its negative side.

Now, given a Hyperplane-Consistency problem we shall reduce it to a single-use mechanism design problem in three steps:

1. We will first show that a hyperplane-consistency problem for vectors with only positive coordinates is still NP-complete;
2. We will then show that using an unbiased hyperplane to separate positive vectors is still just as hard;
3. Finally, we shall reduce the last problem to a mechanism design problem, thus showing that it is NP-Complete to design the mechanism.

Let us assume that we are given a Hyperplane-Consistency problem $(\mathcal{P}, \mathcal{N}, k)$. We now define a new problem Pos-Hyperplane-Consistency as the tuple $(\mathcal{P}', \mathcal{N}', k)$ where

$$\mathcal{P}' = \{\vec{x} + \vec{\delta} \,|\, x \in \mathcal{P}\} \tag{65}$$

$$\mathcal{N}' = \{\vec{x} + \vec{\delta} \,|\, x \in \mathcal{N}\} \tag{66}$$

and where $\vec{\delta}$ has been set large enough in each coordinate to turn all vectors in $\mathcal{P}'$ and $\mathcal{N}'$ positive.

We will now show that given a hyperplane that separates points in a certain way in the original problem, we can build a hyperplane that will separate the matching points in the new problem in the same manner and vice versa. Given a hyperplane $(\vec{w}, b)$, we look at the hyperplane $(\vec{w}, b + \vec{w} \cdot \vec{\delta})$ in the new problem.

$$\vec{w} \cdot \vec{x} \geq b \iff \vec{w} \cdot (\vec{x} + \vec{\delta}) \geq \vec{b} + \vec{w} \cdot \vec{\delta} \tag{67}$$

Meaning that a point $\vec{x}$ and its corresponding point $\vec{x} + \vec{\delta}$ in the new problem are classified in the same way by the hyperplane. In other words, if some hyperplane exists (in either problem) that manages to correctly classify k points or more, then a matching classifier exists in the other problem as well. Pos-Hyperplane-Consistency is therefore also NP-Complete.

Now, we shall show that Pos-Unbiased-Hyperplane-Consistency (where the hyperplanes are unbiased) is still an NP-Complete problem. Given an instance $(\mathcal{P}', \mathcal{N}', k)$ of Pos-Hyperplane-Consistency, we shall reduce it to an instance $(\mathcal{P}'', \mathcal{N}'', k)$ of Pos-Unbiased-Hyperplane-Consistency by adding a coordinate to each vector in $\mathcal{P}'$ and $\mathcal{N}'$:

$$\mathcal{P}'' = \{(\vec{x}, 1) \,|\, x \in \mathcal{P}'\} \tag{68}$$

$$\mathcal{N}'' = \{(\vec{x}, 1) \,|\, x \in \mathcal{N}\} \tag{69}$$

The last coordinate in the vectors now takes the place of the bias. For every hyperplane defined by $\vec{w}, b$, there is a matching unbiased hyperplane with a weight vector $(\vec{w}, -b)$ that gives the same classification to the matching point in the new problem.

$$\vec{w} \cdot \vec{x} \geq b \iff (\vec{w}, -b) \cdot (\vec{x}, 1) \geq 0 \tag{70}$$

A correct classification of k or more points exists in the new problem iff it existed in the old problem, and Pos-Unbiased-Hyperplane-Consistency is NP-Complete as well.

We shall now see the final step in the proof—a reduction from Pos-Unbiased-Hyperplane-Consistency to the mechanism design problem. Given an instance of the former, $(\mathcal{P}'', \mathcal{N}'', k)$ in an n dimensional space, we shall reduce it to the following design problem:

$$\Omega = \{1, \ldots, n+2\} \quad ; \quad X = \{0, 1\} \tag{71}$$

$$S = \mathcal{P}'' \cup \mathcal{N}'' \quad ; \quad \theta = \frac{1}{2}(1 + \frac{k}{|S|}) \tag{72}$$

If $\vec{s} \in \mathcal{P}''$ we define:

$$\vec{p}_{0,\vec{s}} = \frac{\alpha}{||\vec{s}||_1} \cdot (\vec{s}, 0, 0) \tag{73}$$

$$\vec{p}_{1,\vec{s}} = \alpha \cdot (\vec{0}, 1, 0) \tag{74}$$

Otherwise $\vec{s} \in \mathcal{N}''$ and we define:

$$\vec{p}_{0,\vec{s}} = \alpha \cdot (\vec{0}, 0, 1) \tag{75}$$

$$\vec{p}_{1,\vec{s}} = \frac{\alpha}{||\vec{s}||_1} \cdot (\vec{s}, 0, 0) \tag{76}$$

Where α in the above is a normalizing constant that makes sure the probabilities sum to 1. In the above, we assume without loss of generality that no point appears both in $\mathcal{P}''$ and in $\mathcal{N}''$, otherwise it can be eliminated from the problem while reducing k by 1. Now, with the definitions above, a payment mechanism which is simply a vector $\vec{v}_{0,1}$ works for state $x, \vec{s}$ if the vector $\vec{p}_{x,\vec{s}}$ is positioned on the correct side of the hyperplane it represents. The vectors of the form $\alpha \cdot (\vec{0}, 1, 0)$ and $\alpha \cdot (\vec{0}, 0, 1)$ can always be placed on the correct side since they have a coordinate dedicated just to them for that purpose. They constitute one half of the probability weight. The other half are actually vectors identical to the vectors in $\mathcal{P}''$ and in $\mathcal{N}''$ with zeros in their extra coordinates, and a correct separation of k out of them implies directly the correct separation of vectors in the original problem and vice versa. □

References

[1] R. Axelrod, The Evolution of Cooperation, Basic Books, New York, 1984.

[2] E. M. Maximilien, M. P. Singh, Reputation and endorsement for web services, SIGecom Exch. 3 (1) (2002) 24–31.

[3] P. Resnick, K. Kuwabara, R. Zeckhauser, E. Friedman, Reputation systems, Communications of the ACM 43 (12) (2000) 45–48.

[4] A. Mas-Colell, M. D. Whinston, J. R. Green, Microeconomic Theory, Oxford University Press, 1995, Ch. 23, pp. 857–926.

[5] E. Maskin, T. Sjöström, Implementation theory, Working paper, Harvard University and Penn State (January 2001).

[6] J.-J. Laffont, D. Martimort, The Theory of Incentives: The Principal-Agent Model, Princeton University Press, 2002.

[7] B. Salanié, The Economics of Contracts: a Primer, MIT Press, 2005.

[8] J. Pearl, Probabilistic reasoning in intelligent systems: networks of plausible inference, 2nd Edition, Morgan Kaufmann, San Mateo, CA, 1988.

[9] C. Boutilier, On the foundations of expected expected utility, in: The International Joint Conference on Artificial Intelligence (IJCAI'03), Acapulco, 2003, pp. 285–290.

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

[11] T. Gneiting, A. E. Raftery, Strictly proper scoring rules, prediction, and estimation, Tech. Rep. 463, Department of Statistics, University of Washington (2004).

[12] A. D. Hendrickson, R. J. Buehler, Proper scores for probability forecasters, Annals of Mathematical Statistics 42 (1971) 1916–1921.

[13] R. J. Aumann, Agreeing to disagree, The Annals of Statistics 4 (6) (1976) 1236–1239.

[14] D. Samet, Iterated expectations and common priors, Games and Economic Behavior 24 (1).

[15] N. Miller, P. Resnick, R. Zeckhauser, Eliciting honest feedback: The peer prediction method, Management Science 51 (9) (2005) 1359–1373.

[16] R. Jurca, B. Faltings, An incentive compatible reputation mechanism, in: Proceedings of the IEEE Conference on E-Commerce, Newport Beach, CA, USA, 2003, pp. 285–292.

[17] S. Goldwasser, Multi party computations: past and present, in: Proceedings of the Sixteenth Annual ACM Symposium on Principles of Distributed Computing (PODC'97), ACM, New York, NY, USA, 1997, pp. 1–6.

[18] R. Smorodinsky, M. Tennenholtz, Sequential information elicitation in multi-agent systems, in: Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence (UAI-2004), 2004, pp. 528–535.

[19] R. Smorodinsky, M. Tennenholtz, Overcoming free-riding in multi-party computation: the anonymous case, Games and Economic Behavior 55 (2006) 385–406.

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

[21] J. Wolfers, E. Zitzewitz, Prediction markets, Working Paper 10504, National Bureau of Econ. Research (2004).

[22] K. Gajos, D. S. Weld, Preference elicitation for interface optimization, in: UIST'05: Proceedings of the 18th annual ACM symposium on User interface software and technology, ACM Press, New York, NY, USA, 2005, pp. 173–182.

[23] P. Pu, B. Faltings, M. Torrens, User-involved preference elicitation (2003).

[24] J. O. Kephart, J. E. Hanson, J. Sairamesh, Price and niche wars in a free-market economy of software agents, Artificial Life 4 (1) (1998) 1–23.

[25] J. O. Kephart, C. H. Brooks, R. Das, Pricing information bundles in a dynamic environment, in: EC'01: Proceedings of the 3rd ACM conference on Electronic Commerce, ACM Press, New York, NY, USA, 2001, pp. 180–190.

[26] C. Prendergast, A theory of "yes men", The American Economic Review 83 (4) (1993) 757–770.

[27] N. Nisan, A. Ronen, Algorithmic mechanism design (extended abstract), in: STOC'99: Proceedings of the Thirty-First annual ACM Symposium on Theory of Computing, ACM Press, New York, NY, USA, 1999, pp. 129–140.

[28] V. Conitzer, T. Sandholm, Complexity of mechanism design, in: Proceedings of the Uncertainty in Artificial Intelligence Conference (UAI-2002), Edmonton, Canada, 2002, pp. 103–110.

[29] N. Hyafil, C. Boutilier, Partial revelation automated mechanism design, in: The Twenty-Second Conference on Artificial Intelligence (AAAI'07), Vancouver, 2007, pp. 72–78.

[30] N. Hyafil, C. Boutilier, Mechanism design with partial revelation, in: The Twentieth International Joint Conference on Artificial Intelligence (IJCAI'07), Hyderabad, India, 2007, pp. 1333–1340.

[31] R. Jurca, B. Faltings, Eliminating undesired equilibrium points from incentive compatible reputation mechanisms, in: Proceedings of the Seventh International Workshop on Agent Mediated Electronic Commerce (AMEC VII), Utrecht, The Netherlands, 2005.

[32] P. Kall, S. W. Wallace, Stochastic Programming, Systems and Optimization, John Wiley, 1995.

[33] A. Ben-Tal, A. Nemirovski, Robust solutions of uncertain linear programs, Op. Research Letters 25 (1999) 1–13.

[34] J. Håstad, Clique is hard to approximate within $n^{1-\epsilon}$, Acta Mathematica 182 (1999) 105–142.

[35] H. Edelsbrunner, Algorithms in Combinatorial Geometry, Vol. 10 of EATCS Monographs on Theoretical Computer Science, Springer-Verlag, 1987.

[36] M. Vidyasagar, A Theory of Learning and Generalization, Spinger-Verlag, 1997.

[37] A. Zohar, J. S. Rosenschein, Robust mechanisms for information elicitation, in: The Twenty-First National Conference on Artificial Intelligence (AAAI'06), Boston, 2006, pp. 740–745.

[38] A. Zohar, J. S. Rosenschein, Mechanisms for partial information elicitation: The truth, but not the whole truth, in: The Twenty-First National Conference on Artificial Intelligence (AAAI'06), Boston, 2006, pp. 734–739.

[39] E. Amaldi, V. Kann, The complexity and approximability of finding maximum feasible subsystems of linear relations, Theoretical Computer Science 147 (1–2) (1995) 181–210.