Theoretical Economics 13 (2018), 553–578 1555-7561/20180553

Computational principal–agent problems

Pablo D. Azar
Department of Economics, Massachusetts Institute of Technology

Silvio Micali
Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology

Collecting and processing large amounts of data is becoming increasingly crucial in our society. We model this task as evaluating a function f over a large vector $x = (x_1, \ldots, x_n)$, which is unknown, but drawn from a publicly known distribution X. In our model, learning each component of the input x is costly, but computing the output $f(x)$ has zero cost once x is known. We consider the problem of a principal who wishes to delegate the evaluation of f to an agent whose cost of learning any number of components of x is always lower than the corresponding cost of the principal. We prove that, for every continuous function f and every $\epsilon > 0$, the principal can—by learning a single component x_i of x—incentivize the agent to report the correct value $f(x)$ with accuracy ϵ. complexity.

Keywords. Principal agent problems, computational complexity.

JEL classification. D82, D86.

1. Introduction

Our society is more and more data intensive. Every day, firms need to gather and process multiple pieces of data to make products and decisions. In this paper, we investigate how to delegate to a rational agent the process of first obtaining multiple pieces of data and then aggregating them into a "compact" final result. In our model, obtaining the pieces of data is expensive, but the algorithmic operations to process them are free.

In our setting, a risk-neutral principal wants to learn the output of a continuous function $f : [0, 1]^n \to \mathbb{R}$ on an input $x = (x_1, \ldots, x_n)$. The input x is unknown, but drawn from a known distribution X. Once the input vector $(x_1, \ldots, x_n)$ is known, $f(x_1, \ldots, x_n)$ can be computed at no cost. Learning the inputs, however, is costly. Specifically, the principal can learn any k coordinates of x at a cost of $\gamma(k)$. There also exists a risk-neutral agent, who can learn any k coordinates of x at a smaller cost, $c(k) < \gamma(k)$, where both $c(k)$ and $\gamma(k)$ are increasing with k. Since the agent has a lower cost, it is socially optimal for the agent to learn $y = f(x)$ and then report y to the principal. However,

Pablo D. Azar: pazar@mit.edu
Silvio Micali: silvio@csail.mit.edu

We thank Gabriel Carroll, Tommaso Denti, Harry Di Pei, Juuso Toikka, and Vira Semenova for productive discussions as well as the referees and editors for their helpful suggestions. We give special thanks to Georgios Vlachos for his input on the proof of Theorem 2. Pablo D. Azar is grateful for financial support from the Robert Solow Fellowship and the Stanley and Rhoda Fischer Fellowship.

 https://doi.org/10.3982/TE1815

since the principal cannot monitor the agent's actions, she cannot necessarily trust him to learn $f(x)$ or to truthfully report its value. Thus, we must find a way to incentivize the agent to act in the best interests of the principal in this new framework.

Let us illustrate the usefulness of our goals by means of two examples.

EXAMPLE 1 (Scientific Data). Each input x_i is the result of an expensive but replicable physical experiment, which the agent can perform more cheaply than the principal. ◊

EXAMPLE 2 (Proprietary Data). An information company has a proprietary database about the behavior of n individuals, and the principal is an outsider wishing to obtain some aggregate function of these data, say, to price a new product. Here, the agent is the information company itself, and while the principal can recreate each individual's data from scratch with nontrivial cost, the information company—after sinking a fixed cost into creating its database—can retrieve each record very cheaply. ◊

Although our framework and results apply also when the number of inputs $n > 1$ is small and each input component x_i consists of just a few digits, our results are most relevant when n is large and each x_i has a large number of significant digits, so that it would be impractical for an agent to report the entire input vector $(x_1, \ldots, x_n)$ to the principal. Typically, in fact, it is when $(x_1, \ldots, x_n)$ is huge that one wishes to deal instead with an aggregate value $f(x_1, \ldots, x_n)$.[1] Indeed, when the input vector x is large,[2] the principal should not insist on the agent revealing all the data he has learned, but on his reporting the right answer $f(x)$.

We investigate delegation of computation over costly inputs in two settings.

1.1 *Exact computation*

In our first setting, the *exact computation case*, the principal wants to learn $y = f(x_1, \ldots, x_n)$ exactly. We show that, for an important class of functions, the principal can incentivize the agent to reveal $f(x)$ using a direct mechanism. Informally, the following conditions hold in a direct mechanism.

1. The agent (a) chooses a subset S_A of the input coordinates and learns the corresponding subvector $x_{S_A} = (x_i)_{i \in S_A}$ at a cost of $c(|S_A|)$ and (b) reports a value $z(x_{S_A})$, allegedly equal to $y = f(x_1, \ldots, x_n)$.

2. The principal (a) chooses a (random) subset S of the input coordinates and learns $x_S = (x_i)_{i \in S}$ and (b) pays the agent a reward $R(z, x_S)$, which is a continuous function of z and x_S.

[1] As an example, when f is Lipschitz continuous with Lipschitz constant 1, so that $|f(x) - f(x')| < \delta$ when $\|x - x'\|_\infty < \delta$, the number of significant digits that we would need to send to communicate $f(x)$ increases with $\frac{1}{\delta}$, while the number of significant digits needed to communicate the entire vector $(x_1, \ldots, x_n)$ increases at a rate that is n times larger.

[2] For example, the large hadron collider at CERN produces 30 petabytes of data every year (CERN Computing).

A *direct mechanism* is (a) *incentive-compatible* if the agent maximizes his expected payoff $\mathbb{E}_S[R(z, x_S) - c(|S_A|)]$ only by learning $f(x_1, \dots, x_n)$ exactly and reporting $z = f(x_1, \dots, x_n)$ and (b) *individually rational* if the agent's expected reward minus his costs are greater than or equal to zero.

Our first result: The mechanism $\mathcal{M}_1$ A function $f : [0,1]^n \to \mathbb{R}$ is separable if $f(x_1, \dots, x_n) = \sum_{i=1}^n f_i(x_i)$. We show that if f is separable and each f_i is bounded, then the principal can correctly learn $f(x_1, \dots, x_n)$ by means of a direct, incentive-compatible, and individually rational mechanism $\mathcal{M}_1$ in which the principal queries just one input herself.

As we shall see, mechanism $\mathcal{M}_1$ is a crucial subroutine for our more general mechanisms.

Our second result: The limits of direct contracts Our first result raises the question of whether direct contracts can be used to delegate the exact computation of functions that are not separable. In our second theorem, we show that this is not the case even for a very simple nonseparable function. Indeed we show that for

$$f(x) = \prod_{i=1}^n x_i,$$

no direct contract can incentivize the agent to reveal $f(x)$ unless the principal queries the entire input vector $(x_1, \dots, x_n)$ herself.[3]

1.2 *Approximate computation*

In our second setting, the *ϵ-approximate computation case*, the principal wants to learn z such that $|z - f(x)| \leq \epsilon$ for some arbitrarily small $\epsilon > 0$.

Our third result: Approximately delegating arbitrary continuous functions We show that for any continuous function $f : [0,1]^n \to \mathbb{R}$, any input $x \in [0,1]^n$, and any $\epsilon > 0$, there exists a (nondirect) mechanism $\mathcal{M}_2$ that incentivizes the agent to reveal a z that is within distance ϵ of $f(x)$.

In mechanism $\mathcal{M}_2$, the following conditions hold:

- The principal queries only one coordinate of the input vector x.
- The principal and the agent interact in two rounds, in each of which one sends a message to the other.

[3]It is important to remark that our positive results (Theorems 1 and 3) in this paper are proved for functions defined over $[0,1]^n$, whereas the counterexample function $f(x) = \prod_{i=1}^n x_i$ we use to prove our negative results (Theorems 2 and 4) is defined over $[-1,1]^n$. We note that we can always scale and shift this domain and use the function $\tilde{f}(x) = \prod_{i=1}^n (2x_i - 1)$ defined over $[0,1]^n$, but we prefer not to do so to simplify the notation in the proofs of Theorems 2 and 4.

Our fourth result: Round optimality of $\mathcal{M}_2$ In our fourth result, we show that one-round mechanisms cannot be used for approximate delegation of arbitrary continuous functions unless the principal queries the whole input vector $(x_1, \ldots, x_n)$, which would defeat the purpose of delegating the computation in the first place. Accordingly, our mechanism $\mathcal{M}_2$ simultaneously minimizes the number of queries that the principal makes to the input vector, and the number of rounds of interaction between the principal and the agent.

1.3 *Optimality and unlimited liability*

Our mechanisms $\mathcal{M}_1$ minimizes the number of queries and interaction rounds among all incentive-compatible mechanisms that delegate separable functions, and $\mathcal{M}_2$ minimizes the number of queries and interaction rounds among all incentive-compatible mechanisms that approximately delegate continuous functions. These properties hold for any environment where the agent and principal are risk-neutral. In particular, they hold when the agent has limited liability and can only be paid a positive amount ex post.

Of course, queries and rounds of interaction are not the only costs that the principal faces. She must also bear the monetary cost of paying the agent's reward. We highlight that there is one setting where we can simultaneously minimize all three of these costs: namely, when the agent also has unlimited liability.

With unlimited liability, the technique for minimizing the expected payment to the agent is well known (Hölmstrom 1979): the principal charges the agent a fixed participation fee equal to the expected utility that the agent gets from the mechanism.[4] In this way, the principal can always find a mechanism that makes the agent's individual rationality constraint bind, without affecting the number of queries or the number of communication rounds.

Auditing and optimality When the agent has unlimited liability, there is a trivial incentive-compatible, individually rational, and one-round mechanism in which the principal learns $f(x_1, \ldots, x_n)$ exactly, minimizes the expected reward, and makes (in expectation) arbitrarily few queries. Namely, the following mechanism.

The ϵ-Auditing Mechanism

1. The agent reports a value z, allegedly equal to $y = f(x_1, \ldots, x_n)$.
2. With probability $(1 - \epsilon)$, the principal pays the agent 0.
3. With probability ϵ, the principal queries the entire input vector $(x_1, \ldots, x_n)$ and pays the agent $\frac{c(n)}{\epsilon}$ if $f(x_1, \ldots, x_n) = z$ and 0 otherwise.

The ϵ-auditing mechanism, however, is not meaningful in many cases: for instance, when $\gamma(k) = +\infty$ for some $k > 1$. Consider first our Example 1, where each x_i is the

[4]The principal can compute this expected reward at zero cost because she knows the random variable X from which the true input x is drawn and can take expectations with respect to this distribution without gathering any costly information about the true input x. Of course, this would not be true in a model where the principal has to pay a cost for every algorithmic operation that she makes.

result of a replicable scientific experiment. Assume $n = 100$, and that each experiment takes 1 day for the agent to perform, but 1 year for the principal. Then, for every possible choice of ϵ, the agent should never agree to participate in the ϵ-auditing mechanism, as the principal will die before he could pay the agent a positive amount.

Consider now our Example 2, where $(x_1, \dots, x_n)$ is a proprietary data set owned by an information company. If the principal does not have enough money (or time) to learn the entire vector $(x_1, \dots, x_n)$, then the agent has no hope to be paid. (Problems also arise even when the agent is able to reveal and prove to the principal the value of each x_i in a very cheap manner. Indeed, the information company must reveal all of its proprietary data to get paid, and, thus, it destroys its own business by enabling a competitor in the process.)

In any case, even when the ϵ-auditing mechanism can be meaningfully used, we must ask, "what is the optimal mechanism to use once the auditing state is reached?" This is the mechanism that would be observable (when there is no auditing, we only observe that there is no auditing). Thus, we will focus on mechanisms that—under any possible realization of the random choices they make—query at least one input. As we shall prove in Sections 3 and 4, $\mathcal{M}_1$ and $\mathcal{M}_2$ both make only one query to the input distribution and, under unlimited liability, are, therefore, optimal once the auditing state is reached for their respective settings.

1.4 *Additional related work*

Delegating computation to an expert can be interpreted as a moral hazard problem. The agent's effort corresponds to the number of components of x that he queries. The project is successful only if the principal learns $f(x)$ correctly, which requires maximum effort from the agent. Furthermore, the agent has to decide ex ante how much effort to exert before observing any components of x.

There are many results in the proper scoring rule literature that consider costly information acquisition. Osband (1989), Clemen (2002), and Lambert (2013) consider modifications of strictly proper scoring rules where the agent can increase the precision of his signal by paying a cost. These models are like ours because the agent must exert some costly effort to acquire more information, and the decision to expend this effort is made ex ante. In fact, our mechanisms use strictly proper scoring rules as a crucial component to incentivize the agent to acquire information about x and then reveal $f(x)$.

Demski and Sappington (1987) consider a moral hazard model where a (weakly) risk-averse agent must be incentivized to (a) acquire information about a random state of the world s and (b) take an action a that produces some outcome $y = p(s, a)$. The principal is risk-neutral and can only observe the outcome y. Her payoff is given by $y - t(y)$, where t is some transfer to the agent that only depends on the observed outcome y. This model—like ours—captures scenarios where expertise is both costly to acquire and costly to communicate. In fact, in Demski and Sappington's model, communication is infinitely expensive: the principal and agent cannot exchange messages, and the transfer can only depend on the observed outcome. Our paper is similar to Demski and Sappington's model in the sense that the agent learns a function of the state of the world,

but due to communication costs, cannot share his entire expertise with the principal. Instead, he must take an action that depends on his acquired expertise, which produces a low-dimensional outcome that is the only information observable by the principal.

There are many other papers that consider moral hazard with costly information acquisition (see Zermeño Vallés 2012, Carroll 2017, Malcolmson 2009, and the references therein). The main conceptual difference between our results and the moral hazard with costly information acquisition literature is that our model allows both the principal and the agent to acquire information at a cost, with the information asymmetry arising from the fact that the principal's cost for acquiring information is higher than the agent's respective cost. Furthermore, we focus on a problem that has been overlooked so far; that is, how to delegate the evaluation of any continuous function while observing as few of the inputs as possible.

Our model can also be viewed as a way to capture the complexity of contracts. One other way to account for computation in the design of contracts was studied by Anderlini and Felli (1994), who show that when contracts are generated by a Turing machine, then in some situations computable contracts will be suboptimal. In subsequent work, Al-Najjar et al. (2006) show that when events cannot be finitely described, sometimes the best contract is no contract at all. Since our model focuses on the computation of continuous functions, where the complexity of a contract depends only on the number of inputs queried, we bypass these impossibility results. Another way in which complexity in contracts can be modeled is via how much time it takes to resolve uncertainty in the state of the world. Using this, MacLeod studies what types of contracts are more efficient in low and high complexity environments (MacLeod 2000).

In a previous paper of ours Azar and Micali (2012), we study the problem of delegating the computation of a function, where both principal and agent have zero cost for computing, but the principal cannot perform more than a given amount of computation. In contrast, the model we present in this paper captures the fact that computation is costly for both the principal and the agent, and that differences in cost should capture realistic differences in computational power.

Furthermore, our earlier results Azar and Micali (2012) and followup work by Guo et al. (2015) focus on giving alternative characterizations of computational complexity classes as sets of problems that can be delegated to a rational agent. In particular, when the agent is computationally unlimited, the set of problems that can be delegated to a computationally unbounded and rational agent in one round of communication includes the complexity class $\#P$ (Azar and Micali 2012). This is a set of canonical counting problems that includes counting the number of satisfying assignments to a boolean formula, and counting the number of perfect matchings in a bipartite graph. When the principal is restricted to act in sublinear time and the agent is restricted to act in polynomial time (but both still have zero cost of computation), the set of problems that the principal can delegate to the agent is the complexity class P (Guo et al. 2015). This is the set of problems that can be solved in polynomial time.

We remark that—like our results in this paper—the results in Azar and Micali (2012) and Guo et al. (2015) use proper scoring rules to incentivize the agent to compute a large sum in a truthful manner.

2. The model

Notation

We denote the set $\{1,\dots,n\}$ by $[n]$. For any $S \subset [n]$, we denote by x_S the vector $(x_i)_{i\in S}$ and by x_{-S} the vector $(x_i)_{i\notin S}$. We emphasize that knowledge of x_S implies knowledge of the underlying coordinate set S. We refer to each subvector x_S as a partial input.

If $X = (X_1,\dots,X_n)$ is a random variable over $\mathbb{R}^n$, we define $X_S \stackrel{\text{def}}{=} (X_i)_{i\in S}$ and denote the conditional random variable "X given that $X_S = x_S$" by $X|x_S$. We refer to the process of learning x_S as *querying* the subset S.

If g is a function of a random variable X, we will write the expectation of g as $\mathbb{E}_{x\leftarrow X}[g(x)]$. We write the expectation of a function $g(\cdot,\cdot)$ with respect to two independent random variables X and Y as $\mathbb{E}_{x\leftarrow X, y\leftarrow Y}[g(x,y)]$.

Computational environments

A *computational environment* is a tuple, $\mathcal{E} = (f, X, x, c, \gamma)$, where the following definitions hold:

- The function $f : [0,1]^n \to \mathbb{R}$ is the *target function*.
- The random variable $X \in \Delta([0,1]^n)$ is continuously distributed, with a distribution that is common knowledge to the principal and the agent.
- The vector $x = (x_1,\dots,x_n)$ is a realization of the random variable X, a priori unknown to the principal or agent.
- The function $c : \{0,\dots,n\} \to \mathbb{R}$ is the *agent's cost function*. For every set $S \subset [n]$, $c(|S|)$ is the cost to the agent of learning x_S.
- The function $\gamma : \{0,\dots,n\} \to \mathbb{R}$ is the *principal's cost function*. For every set $S \subset [n]$, $\gamma(|S|)$ is the cost to the principal of learning x_S.

Throughout the paper, we maintain the following assumptions.

Assumption 0. Any sequence of purely algorithmic operations has zero cost for both the principal and agent. (Thus, evaluating any function on known inputs, or computing an expectation over a known distribution, can be performed at zero cost.)

Assumption 1. The cost functions are monotonic. For all $\ell < k$, we have

$$\gamma(\ell) \le \gamma(k), \qquad c(\ell) \le c\big(k\big).$$

Assumption 2. The cost of querying zero inputs is zero:

$$\gamma(0) = c(0) = 0.$$

Assumption 3. The cost functions satisfy the increasing differences condition. For all ℓ, k, we have

$$c(k+\ell) - c(k) \le \gamma(k+\ell) - \gamma(k).$$

(That is, it is always cheaper for the agent than for the principal to query ℓ extra inputs.)

3. Exact computation

In this section, we prove our results for the exact computation setting. Before stating our results, we define direct computational contracts, define boundedly separable functions, and recall some facts about proper scoring rules.

3.1 *Preliminaries*

Definition 1. A set $S \subset [n]$ *determines* f *exactly* if the support of the random variable $f(X)|x_S$ is a singleton for every possible value of x_S.

Definition 2. For all $k \leq n$, a k*-query direct computational contract* is a mechanism $\mathcal{M}$ specified by the following statements:

- A function $\mathcal{D}$ mapping a real number z to a distribution $\mathcal{D}(z)$ over $\{S \subset [n] : |S| = k\}$, that is, over all input coordinate subsets of size k.
- A continuous reward function R mapping a real number z, and a partial input x_S to a real number $R(z, x_S)$.

Such a mechanism $\mathcal{M} = (\mathcal{D}, R)$ has only one player, the agent, and is played as follows.

Stage 0. Nature draws $x \leftarrow X$. (The agent does not observe x.)

Stage 1. The agent queries a subset S_A of the inputs (updating his beliefs about x to $X|x_{S_A}$).

Stage 2. The agent reports a value $z = z(x_{S_A})$ to the mechanism.

Stage 3. The mechanism draws a subset S of inputs from the distribution $\mathcal{D}(z)$.

In such a play, the agent's reward is $r = R(z, x_S)$ and his utility is $r - c(|S_A|)$.

The mechanism $\mathcal{M}$ is *incentive-compatible* for a function $f : [0,1]^n \to \mathbb{R}$ if the agent strictly maximizes his utility by choosing S_A such that S_A determines f exactly, and reporting $z = f(x)$ in Stage 2.

The mechanism $\mathcal{M}$ is *individually rational* for f if $\mathbb{E}_{S \leftarrow \mathcal{D}(f(x)), x \leftarrow X}[R(f(x), x_S) - c(n)] \geq 0$.

Definition 3. A function $f : [0,1]^n \to \mathbb{R}$ is *boundedly separable* if there exist bounded functions $f_1, \ldots, f_n : [0,1] \to \mathbb{R}$ such that $f(x_1, \ldots, x_n) = \sum_{i=1}^n f_i(x_i)$.

Scoring rules A *strictly proper scoring rule*[5] is a function $S : \Delta(K) \times K \to \mathbb{R}$ that takes as input a distribution Ω over a finite set K and a sample $\omega \in K$, and that satisfies the

[5]For brevity and when the context is clear, we will often refer to strictly proper scoring rules simply as proper scoring rules or scoring rules.

property

$$\mathop{\mathbb{E}}_{\omega\leftarrow\Omega} S(\Omega,\omega) > \mathop{\mathbb{E}}_{\omega\leftarrow\Omega} S\big(\Omega',\omega\big)$$

for any $\Omega' \neq \Omega$. A function that satisfies this property incentivizes a rational expert to state his true beliefs about the distribution of ω. Many such rules are known.[6] We will use Brier's scoring rule (BSR), defined by

$$\mathrm{BSR}(\Omega,\omega) = 2\Pr(\Omega=\omega) - \sum_{\alpha}\Pr(\Omega=\alpha)^2 + 1,$$

where $\Pr(\Omega=\alpha)$ is the probability that distribution Ω assigns to the element α. The Brier scoring rule is always in the interval $[0,3]$.[7]

3.2 *Our first theorem*

THEOREM 1. *Let $f:[0,1]^n \to \mathbb{R}$ be a boundedly separable function. Then there exists a one-query, incentive-compatible, and individually rational direct computational contract $\mathcal{M}_1 = (R,\mathcal{D})$ for f.*

PROOF. We first prove Theorem 1 assuming that the agent has zero costs; that is, $c(|S_A|) = 0$ for all $S_A \subset [n]$. Let $f(x_1,\dots,x_n) = f_1(x_1) + \cdots + f_n(x_n)$ and let $B > 0$ be a bound such that $|f_i(x)| < B$ for every $x \in [0,1]$ and every $i \in [n]$. Let $g_i(x) = f_i(x_i) + B$ and note that $0 \le g_i(x) \le 2B$. Let $g(x_1,\dots,x_n) = \sum_{i=1}^n g_i(x) = f(x_1,\dots,x_n) + nB$. We now give an incentive-compatible contract for g. Since, by construction, we have ensured that each term $g_i(x)$ in the sum is nonnegative, our mechanism can scale this term so as to interpret it as a probability, and use some techniques from scoring rules to incentivize the agent.

Our mechanism $\mathcal{M}_1$ takes the agent's report z as an input and produces a distribution $\mathcal{D}(z)$ and reward function $R(z,\cdot)$ as follows:

$$\text{Mechanism } \mathcal{M}_1(z) = \big(\mathcal{D}(z), R(z,\cdot)\big)$$

- The function $\mathcal{D}(z)$ is the uniform distribution over the singleton sets $\{\{1\},\dots,\{n\}\}$.
- The reward function $R(z,\cdot)$ is defined as follows:
  - If $z \notin [-Bn, Bn]$, then $R_1(z, x_S) = -\infty$ for all x_S. (That is, since the range of f is $[-Bn, Bn]$, a z outside this range must be a lie.)
  - If $z \in [-Bn, Bn]$, then the mechanism proceeds as follows:
    - Draws $S = \{i\}$ from $\mathcal{D}$. Since S is a singleton, we denote it by $S = i$.

[6] The interested reader is referred to a paper by Gneiting and Raftery (2007), which includes a comprehensive survey.

[7] Usually, the Brier scoring rule is defined as $\mathrm{BSR}(\Omega,\omega) = 2\Pr(\Omega=\omega) - \sum_\alpha \Pr(\Omega=\alpha)^2 - 1$. Our formula is the usual definition plus 2. The reason we add 2 to the usual formulation of the scoring rule is to ensure that the reward to the expert is nonnegative. Note that adding a constant to this reward does not affect incentives.

- Queries x_i and computes $g_i(x_i)$.
- Draws a realization from the random variable ω that is equal to 1 with probability $\frac{g_i(x_i)}{2B}$ and equal to 0 with probability $1-\frac{g_i(x_i)}{2B}$.
- Interprets z as a random variable Ω_z over $\{0,1\}$ that is equal to 1 with probability $\frac{(z+Bn)}{2Bn}$ and equal to 0 with probability $1-\frac{(z+Bn)}{2Bn}$.
- Returns $R_1(z,x_i)=\mathrm{BSR}(\Omega_z,\omega)$.

Let us now show that mechanism $\mathcal{M}_1$ is incentive-compatible. Given a report z, the agent's expected reward is $\mathbb{E}_i R(z,x_i)=\mathbb{E}_\omega \mathrm{BSR}(\Omega_z,\omega)$. Since Brier's scoring rule is strictly proper, this expected reward is maximized only when the agent reports z so that Ω_z is equal to the distribution from which the principal is drawing ω.

Let us now consider the probability that ω is equal to 1. This probability is equal to $\sum_{i=1}^n \Pr(\omega=1|S=i)\Pr(S=i)$. Since $\Pr(\omega=1|S=i)=\frac{g_i(x_i)}{2B}$ and $\Pr(S=i)=\frac{1}{n}$, we have that

$$\Pr(\omega=1)=\sum_{i=1}^n \Pr(\omega=1|S=i)\Pr(S=i)=\frac{1}{2Bn}\sum_{i=1}^n g_i(x_i).$$

Since Ω_z is a random variable with $\Pr(\Omega_z=1)=\frac{z+Bn}{2Bn}$, the agent maximizes his reward by announcing z such that $z+Bn=\sum_{i=1}^n g_i(x_i)$. Note that this is equivalent to announcing $z=\sum_i f_i(x_i)$, because each $g_i(x_i)=f_i(x_i)+B$. Thus, the agent maximizes his reward by announcing $z=f(x_1,\ldots,x_n)$. Since we are currently assuming that the agent's cost function is identically 0, the agent can learn the value of $f((x_1,\ldots,x_n))$ at no cost by querying $S=[n]$.

The above argument only applies when the agent's cost function c is identically 0. When this is not the case, we can scale the reward $R(z,x)$ by a large enough constant so that the agent is incentivized to learn the value of $f(x_1,\ldots,x_n)$ exactly. This part of our proof is standard in the costly information acquisition literature (Lambert 2013, Osband 1989, Clemen 2002) and proceeds as follows.

For every partial input x_{S_A}, let

$$z^*(x_{S_A})\in \arg\max_z \mathop{\mathbb{E}}_{i\leftarrow\mathcal{D}(z),x\leftarrow X|x_{S_A}}\big[R(z,x_i)\big],$$

$$\nu(S_A)=\mathop{\mathbb{E}}_{i\leftarrow\mathcal{D}(z),x\leftarrow X}\big[R\big(z^*(x_{S_A}),x_i\big)\big],$$

so that $\nu(S_A)$ is the reward that the agent ultimately expects to receive when he chooses to query set S_A. Let $S_A^*\in\arg\max_S \nu(S)$. Since the agent maximizes his reward only if he reports $z=f(x)$ exactly, the random variable $f(X)|x_{S_A}$ must satisfy $\Pr(f(X)=z|x_{S_A})=1$ for any possible realization of x_{S_A}. This implies that the agent learns $f(x)$ exactly when he queries the set S_A^*.[8]

[8]Equivalently, for any x in the support of X, the value of the function $f(x)$ must not depend on $x_{-S_A^*}$.

Depending on the distribution of the random variable X and the choice of f, there might be multiple sets in $\arg\max_S \nu(S)$. Let S_A^* be a set in $\arg\max_S \nu(S)$ that has minimum cardinality.[9] Let $\kappa > 0$ be such that, for any S such that $S_A \notin \arg\max_S \nu(S)$, we have

$$\kappa \cdot \nu(S_A^*) - c(|S_A^*|) > \kappa \cdot \nu(S_A) - c(|S_A|). \tag{1}$$

Since $\nu(S_A)$ can be computed by taking expectations over the commonly known distribution X, and without the need to learn the true input x, the principal can compute κ at zero cost. Let $\tilde{R}(\cdot,\cdot) \stackrel{\text{def}}{=} \kappa \cdot R(\cdot,\cdot)$ be a scaled reward function, and let $\tilde{\nu}(S_A)$ be the reward that the agent expects to receive when he chooses to query S_A and the reward function is $\tilde{R}$. Then, by construction, we have

$$\tilde{\nu}(S_A^*) - c(|S_A^*|) > \tilde{\nu}(S_A) - c(|S_A|)$$

for any $S_A \notin \arg\max_S \nu(S)$ and any $S_A \in \arg\max_S \nu(S_A)$ such that $c(|S_A|) > c(|S_A^*|)$. Thus, by changing the reward function of $\mathcal{M}_1$ to be $\kappa \cdot R$, we can incentivize the agent to learn $f(x_1,\dots,x_n)$ exactly at the minimum cost, and to report $z = f(x_1,\dots,x_n)$.

Finally, since $\nu(S_A^*)$ is positive, the principal can choose κ so that inequality (1) holds, and also $\kappa \cdot \nu(S_A^*) - c(|S_A^*|) \geq 0$. That is, so that $\mathcal{M}_1$ using the reward function $\kappa \cdot R$ is also individually rational. □

3.3 *Query optimality and our second theorem*

Since the mechanism $\mathcal{M}_1$ only makes one query, we have the following corollary.

COROLLARY 1. *No direct computational contract that is incentive-compatible for separable functions makes fewer queries than* $\mathcal{M}_1$.

We now argue that the query optimality of mechanism $\mathcal{M}_1$ is intrinsically linked to the fact that f is separable. Indeed, we show the following theorem.

THEOREM 2. *There exists a continuous, bounded, and nonseparable function* $f : [-1, 1]^n \to \mathbb{R}$ *such that any direct computational contract* $\mathcal{M} = (\mathcal{D}, R)$ *that is incentive-compatible for* f *must query the whole input vector* $(x_1,\dots,x_n)$.

Theorem 2, which is proved in Appendix A, provides the motivation to broaden either our definition of computational contracts or the notion of delegation itself.

In the next section, we define multi-round contracts and show that they can be used to delegate any continuous function approximately. That is, the agent can be incentivized, for all continuous f and all $x \in \mathbb{R}^n$, to reveal some value $z \in [f(x) - \epsilon, f(x) + \epsilon]$, where $\epsilon > 0$ is an arbitrary small number.

[9] Note that the agent is incentivized to query a set with minimum cardinality. For any two sets $S_A^*, S_A^{**} \in \arg\max_{S_A} \nu(S_A)$ such that $|S_A^*| < |S_A^{**}|$, the agent will always prefer to query S_A^* than to query S_A^{**}.

The fact that we broaden both the definition of a contract and the notion of exact delegation raises the question of whether multi-round mechanisms are really necessary for the approximate delegation of an arbitrary continuous function. In Section 5, we prove a generalization of Theorem 2 showing that multi-round mechanisms are necessary. That is, even if we allow approximate answers, we cannot incentivize the agent to reveal an approximate value to $f(x)$ with a one-round mechanism.

4. Approximate computation

Direct computational contracts are sufficient for delegating boundedly separable functions, but to handle the delegation of arbitrary continuous functions in an approximate manner, we need to define computational contract more generally, so as to allow the agent to report not only (an approximation to) the true answer $f(x)$, but some additional evidence that helps the principal compute the reward. In addition, before proving our theorems, we recall a result of Kolmogorov about representing continuous functions, as well as a basic fact about Brier's scoring rule.

4.1 *Preliminaries*

Definition 4. A *k-query, T-round computational contract* for a function $f : [0,1]^n \to \mathbb{R}$ is a mechanism $\mathcal{M}$ specified as follows:

- A collection of functions $\{\mathcal{D}_t\}_{t=1}^T$, where $\mathcal{D}_t : \mathbb{R}^{2(t-1)+1} \to \Delta(D)$ maps a vector of length $2(t-1)+1$ to a distribution $\mathcal{D}_t(m)$ over a finite support D.
- A function $\mathcal{S} : \mathbb{R}^{2T} \to [n]$ that maps a vector of size $2T$ to a set of input coordinates $\mathcal{S}(m)$ of size k.
- A continuous reward function $R : \mathbb{R}^{2T} \times \bigcup_{\ell=1}^n \mathbb{R}^\ell$ that maps a vector of size $2T$ and a partial input x_S to a real number $R(m, x_S)$.

Such a mechanism $\mathcal{M} = (\{\mathcal{D}_t\}_{t=1}^T, \mathcal{S}, R)$ has a single player—the agent—and it is played over T rounds. In each round, the agent sends a message to the mechanism and then receives a random message from the mechanism.

At any round $t \in \{1, \dots, T\}$, the information available to the agent consists of the set S_A^t of all inputs he has queried so far, and of the vector m^t of all messages exchanged with the mechanism so far. Initially, $S_A^0 = \varnothing$ and m^0 is the empty vector. A play of the mechanism proceeds as follows.

Stage 0. Nature draws $x \leftarrow X$. The agent does not observe x.

Round t. For each $t \in \{1, \dots, T\}$, round t consists of the following stages:

Stage $2(t-1)+1$. The agent

- chooses a function $a_t(\cdot)$ and a subset $S_{A,t} \subset [n]$
- queries the set $S_{A,t}$ and updates $S_A^t = S_A^{t-1} \cup S_{A,t}$
- sends the mechanism the message $m_t = a_t(x_{S_A^t})$.

Stage $2(t-1)+2$. The mechanism draws a random element r_t from the distribution $\mathcal{D}_t(m_1, r_1, \ldots, r_{t-1}, m_t)$. The vector of messages at the end of this round is $m^t = (m_1, r_1, \ldots, m_t, r_t)$.

At the end of this play, the mechanism queries the set $S = \mathcal{S}(m^T)$ and pays the agent the reward $r = R(m^T, x_S)$. The agent's utility is $r - c(|S_A^T|)$.

The mechanism $\mathcal{M}$ is *ϵ-incentive-compatible* for f if

$$\big|m_1 - f(x)\big| \le \epsilon,$$

where the sequence of messages $m_1, \ldots, m_T$ is generated by an agent that, at each round $t \in \{1, \ldots, T\}$, chooses $a_t(\cdot)$ and $S_{A,t}$ to maximize his expected utility given the information $x_{S_A^{t-1}}$, m^{t-1} available to him at the beginning of the round.

The mechanism $\mathcal{M}$ is *individually rational* if

$$\max_{a_1(\cdot), S_{A,1}} \mathop{\mathbb{E}}_{x \leftarrow x, r_1 \leftarrow \mathcal{D}_1} \cdots \max_{a_T(\cdot), S_A^T} \mathop{\mathbb{E}}_{x \leftarrow X | (x_{S_A^{T-1}}, m^{T-1}), r_T \leftarrow \mathcal{D}_T} \big[R\big((m_1, r_1, \ldots, m_T, r_T), x_{\mathcal{S}(m)}\big)\big]$$
$$- c\big(\big|S_A^T\big|\big) > 0.$$

Kolmogorov's superposition theorem In Theorem 3, we will make use of the following representation of continuous functions over compact sets.

Theorem (Kolmogorov 1963). *Let $f : [0,1]^n \to \mathbb{R}$ be an arbitrary continuous function. Then f has the representation*

$$f(x) = \sum_{q=0}^{2n} \Phi_q \left(\sum_{p=1}^{n} \psi_{q,p}(x_p) \right),$$

where Φ_q and $\psi_{q,p}$ are continuous one-dimensional functions, and the functions $\psi_{q,p}$ are Lipschitz continuous and independent of the function f.

A basic property of Brier's scoring rule In our proof of Theorem 3, we will use the following well known property of Brier's scoring rule on binary distributions

Lemma 1. *Let v and w be real numbers in $[0,1]$, and let V and W be random variables over $\{0,1\}$ such that $\Pr(V=1) = v$ and $\Pr(W=1) = w$. Then*

$$\mathop{\mathbb{E}}_{\omega \leftarrow V} \big[\mathrm{BSR}(V, \omega) - \mathrm{BSR}(W, \omega)\big] = 2(v-w)^2.$$

For completeness, the proof is given in Appendix B.

4.2 *Our third theorem*

Theorem 3. *For all continuous functions $f : [0,1]^n \to \mathbb{R}$ and all $\epsilon > 0$, there exists a one-query, two-round, computational contract $\mathcal{M}_2$ that is ϵ-incentive-compatible and individually rational.*

PROOF. As in the proof of Theorem 1, we first prove Theorem 3 assuming that the agent's cost function is identically zero. We then use this result to prove the more general result when the agent's cost function is arbitrary.

Case 1: $c(\cdot) \equiv 0$. Let $f(x_1, \ldots, x_n) = \sum_{q=0}^{2n} \Phi_q(\sum_{p=1}^{n} \psi_{q,p}(x_p))$. Since the functions $\psi_{q,p}$ are Lipschitz continuous, there exists an M such that $|\psi_{q,p}(x) - \psi_{q,p}(x')| < M|x - x'|$ for any $x, x' \in [0,1]$. Furthermore, because of this Lipschitz condition, the family of functions $\{\psi_{q,p}\}_{q,p}$ has the following two properties:

- Uniform boundedness. There exists a constant $B > 0$ such that $|\psi_{q,p}(x)| \leq B$ for every $x \in [0,1]$ and every q, p.
- Uniform equicontinuity. For every $\epsilon > 0$, there exists a $\delta > 0$ such that $|x - x'| < \delta$ implies $|\psi_{q,p}(x) - \psi_{q,p}(x')| < \epsilon$ for every q, p and every $x, x' \in [0,1]$.

Because of uniform boundedness, we can interpret the domain of the "outer" function Φ_q as the compact set $[-nB, nB]$. Since any continuous function with a compact domain is bounded and uniformly continuous, we have that each Φ_q is bounded and uniformly continuous. Let C be a bound such that the image of each Φ_q is contained in $[-C, C]$.

INTUITION The intuition behind our contract is to interpret $f(x) = \sum_{q=0}^{2n} \Phi_q \times (\sum_{p=1}^{n} \psi_{q,p}(x_p))$ as a boundedly separable function $\tilde{f}(w_0, \ldots, w_{2n}) = \sum_{q=0}^{2n} \Phi_q(w_q)$, where each $w_q(x) = \sum_{p=1}^{n} \psi_{q,p}(x_p)$ is itself a function of x. If the principal knew the value w_q for a random index q, then she could use the computational contract from Theorem 1 to incentivize the agent to reveal $f(x) = \tilde{f}(w) = \sum_{q=0}^{2n} \Phi_q(w_q)$ using the mechanism $\mathcal{M}_1$.

However, the principal does not know the value of $w_q(x)$. Since $w_q(x)$ is itself a boundedly separable function of x, the principal might attempt the following mechanism.

Round 1. Use mechanism $\mathcal{M}_1$ to incentivize the agent to reveal $\tilde{f}(w(x)) = \sum_{q=0}^{2n} \Phi_q \times (w_q(x))$. This mechanism needs to query $w_q(x)$ for a uniformly random q. To obtain $w_q(x)$, go to round 2.

Round 2. Use mechanism $\mathcal{M}_1$ to incentivize the agent to reveal the boundedly separable function $w_q(x) = \sum_{p=1}^{n} \psi_{q,p}(x_p)$ by querying a uniformly random input coordinate p.

The problem with this approach is that the agent gets two rewards: one for announcing $\tilde{f}(w)$ and one for announcing $w_q(x)$. Accordingly, it is possible that the agent would lie about w_q (thus, getting a lower reward in round 2) to manipulate the mechanism in round 1 and receive a higher reward overall.

The way to avoid this problem is to make the reward from round 2 so high that the agent has no incentive in round 2 to reveal a value v_q whose distance from $w_q(x)$ is greater than δ for some δ that we will choose. We will argue that the agent will be incentivized in round 1 to announce $\tilde{f}(v_1, \ldots, v_n) = \sum_{q=0}^{2n} \Phi_q(v_q)$ instead of the true value

$\tilde{f}(w_1, \ldots, w_n) = \sum_{q=0}^{2n} \Phi_q(w_q)$. Nevertheless, by using the uniform continuity of $\tilde{f}$, we will guarantee that the agent's announcement is ϵ-close to the true value $\tilde{f}(w) = f(x)$.

THE CONTRACT $\mathcal{M}_2$ We formalize the above intuition via the following contract and analysis.

Mechanism $\mathcal{M}_2 = (\mathcal{D}, \mathcal{S}, R)$

A play of $\mathcal{M}_2$ proceeds as follows:

- In Stage 1, the agent announces z, allegedly in the set $[f(x) - \epsilon, f(x) + \epsilon]$.
- In Stage 2, the mechanism announces q, drawn from $\mathcal{D}_1 = \text{Uniform}(\{0, \ldots, 2n\})$.
- In Stage 3, the agent announces v_q, allegedly close to $\sum_{p=1}^{n} \psi_{p,q}(x_p)$.
- In Stage 4, the mechanism announces p, drawn from $\mathcal{D}_2 = \text{Uniform}(\{1, \ldots, n\})$.

Given the message vector $m = (z, q, v_q, p)$, define

- $\mathcal{S}(m) = \{p\} \subset [n]$, and
- $R(m, x_{\mathcal{S}(m)})$, the value computed as follows:
  - Let $\omega_1(v_q)$ be a realization of a random variable that is equal to 1 with probability $\frac{(\Phi_q(v_q)+C)}{2C}$ and equal to 0 with probability $1 - \frac{(\Phi_q(v_q)+C)}{2C}$.
  - Let ω_2 be a realization of a random variable that is equal to 1 with probability $\frac{\psi_{p,q}(x_p)}{2B}$ and equal to 0 with probability $1 - \frac{\psi_{p,q}(x_p)}{2B}$.
  - Interpret z as a random variable Ω_z that is equal to 1 with probability $\frac{(z+(2n+1)C)}{((2n+1)2C)}$ and equal to 0 with probability $1 - \frac{(z+(2n+1)C)}{((2n+1)2C)}$.
  - Interpret v_q as a random variable Ω_{v_q} that is equal to 1 with probability $\frac{(v_q+nB)}{2Bn}$ and equal to 0 with probability $1 - \frac{(v_q+nB)}{2Bn}$.
  - Return the reward

$$R(m, x_{\mathcal{S}(m)}) = R\big((z, q, v_q, p), x_p\big) = \text{BSR}(\Omega_z, \omega_1) + \theta \cdot \text{BSR}(\Omega_{v_q}, \omega_2),$$

where $\theta > 0$ is a constant, which we determine later.

ϵ-INCENTIVE COMPATIBILITY The reward function depends on z only through $\text{BSR}(\Omega_z, \omega_1(v_q))$. Thus, taking his choice of v_q in Stage 3 as given, the agent is incentivized to reveal z in Stage 1 such that the distribution Ω_z equals the distribution from which ω_1 is drawn. We have that

$$\Pr(\omega_1 = 1) = \sum_{q=0}^{2n} \Pr(q) \Pr(\omega_1 = 1 | q) = \frac{1}{2n+1} \sum_{q=0}^{2n} \frac{\Phi_q(v_q) + C}{2C}$$

$$= \frac{1}{(2n+1)2C} \left(\sum_{q=0}^{2n} \Phi_q(v_q) + (2n+1)C \right).$$

Since $\Pr(\Omega_z = 1) = \frac{(z+(2n+1)C)}{((2n+1)2C)}$, we conclude that $\Pr(\Omega_z = 1) = \Pr(\omega_1 = 1)$ if and only if

$$z = \sum_{q=0}^{2n} \Phi_q(v_q).$$

Note that z is not necessarily the true value $f(x_1, \ldots, x_n) = \sum_{q=0}^{2n} \Phi_q(\sum_{p=1}^{n} \psi_{p,q}(x_p))$, since $v_q \neq \sum_{p=1}^{n} \psi_{p,q}(x_p)$. Note further that the agent *is not incentivized* to announce $v_q = \sum_{p=1}^{n} \psi_{p,q}(x_p)$, since v_q enters his reward both via Ω_{v_q} and via ω_1, which is defined in terms of v_q. While announcing $v_q \neq \sum_{p=1}^{n} \psi_{p,q}(x_p)$ always decreases the term $\theta \operatorname{BSR}(\Omega_{v_q}, \omega_2)$ in the agent's reward, it may increase the term $\operatorname{BSR}(\Omega_z, \omega_1)$ enough to make such a deviation profitable.

We now give a bound on how much the agent can profit by announcing v_q instead of $w_q = \sum_{p=1}^{n} \psi_{p,q}(x_p)$. There are two effects that this deviation has on the expected reward:

- The first term of the expected reward changes by $\mathbb{E}_{\omega_1}[\operatorname{BSR}(\Omega_z, \omega_1(v_q)) - \operatorname{BSR}(\Omega_z, \omega_1(w_q)] \leq 3$ since the Brier scoring rule is bounded above by 3 and below by 0.
- Given a fixed q, by Lemma 1, the second term of the expected reward changes by

$$\theta \cdot \mathop{\mathbb{E}}_{\omega_2}\left[\operatorname{BSR}(\Omega_{v_q}, \omega_2) - \operatorname{BSR}(\Omega_{w_q}, \omega_2)\right] = -\frac{2\theta}{(2Bn)^2}(v_q - w_q)^2.$$

  Taking expectations over q, which is drawn uniformly from $\{0, 1, \ldots, 2n\}$, the total expected loss in reward is $-\frac{2\theta}{(2Bn)^2}\frac{1}{(2n+1)}\|v - w\|_2^2$.

This analysis shows that when the agent reports v such that $\|v - w\|_2^2 > \delta$, the change in his reward is bounded by

$$3 - \frac{2\theta}{(2n+1)(2Bn)^2}\delta.$$

When θ is high enough, this change in reward is always negative and the agent always prefers not to announce a value v far away from w. In particular, if we set

$$\theta > \frac{3(2n+1)(2Bn)^2}{2\delta},$$

the agent is incentivized to report v such that $\|v - w\|_2^2 \leq \delta$.

Finally, we use the above analysis to show that the principal can incentivize the agent to announce z such that $|z - f(x)| < \epsilon$. For the desired approximation factor ϵ, let $\delta(\epsilon)$ be such that when $\|v - w\|_2^2 < \delta(\epsilon)$, we have $|\sum_{q=0}^{2n} \Phi_q(v_q) - \sum_{q=0}^{2n} \Phi_q(w_q)| < \epsilon$.[10] Since the agent is incentivized to announce v such that $\|v - w\|_2^2 < \delta(\epsilon)$ when $\theta > \frac{(3(2Bn)^2}{2\delta(\epsilon))}$ and he is always incentivized to announce $z = \sum_{q=0}^{2n} \Phi_q(v_q)$, we conclude that the agent is always incentivized to announce z such that $|z - f(x_1, \ldots, x_n)| < \epsilon$.

[10] Note that $\delta(\epsilon)$ does not depend on x since $\sum_{q=0}^{2n} \Phi_q(\cdot)$ is uniformly continuous.

Case 2: $\mathbf{c}(\cdot) \not\equiv \mathbf{0}$. The above argument only applies when the agent's cost function c is identically 0. When this is not the case, we prove that we can scale the reward $R((z, q, v_q, p), x_p)$ by a large enough constant κ so that the agent is incentivized to learn enough inputs from the vector $(x_1, \ldots, x_n)$ so as to provide an ϵ-approximation to f. As mentioned above, once the agent has learned this ϵ-approximation, the scoring rule guarantees that he will maximize his reward by reporting it to the mechanism. In contrast to the proof of Theorem 1, the following scenarios occur:

- The agent does not need to learn the whole input $(x_1, \ldots, x_n)$, since he only needs to give an approximation to $f(x)$.
- The agent learns his set of inputs S_A in two stages: by querying $S_{A,1}$ in Round 1 of the mechanism and by querying $S_{A,2}(z, q, x_{S_{A,1}})$ in Round 2, after making his first-round announcement z, and learning the mechanism's random choice q and the partial input $x_{S_{A,1}}$. The final set of inputs S_A that the agent learns is the union $S_{A,1} \cup S_{A,2}$.

Our argument proceeds by analyzing the agent's optimization using backward induction. For any partial inputs $x_{S_{A,1}}$ and $x_{S_{A,2}}$ that the agent could learn, any z that the agent announces in Stage 1, and any random q that the mechanism could draw in Stage 2, let

$$v_q^*(z, x_{S_{A,1}}, x_{S_{A,2}}) \in \arg\max_v \mathop{\mathbb{E}}_{x \leftarrow X|(x_{S_{A,1}}, x_{S_{A,2}}), p \leftarrow \mathcal{D}_2} \big[R\big((z, q, v, p), x_p\big)\big],$$

$$\nu_2(z, q, x_{S_{A,1}}, S_{A,2}) = \mathop{\mathbb{E}}_{x \leftarrow X|x_{S_{A,1}}, p \leftarrow \mathcal{D}_2} \big[R\big(\big(z, q, v_q^*(x_{S_{A,1}}, q, x_{S_{A,2}}), p\big), x_p\big)\big],$$

so that $\nu_2(z, q, x_{S_{A,1}}, S_{A,2})$ is the reward that the agent ultimately expects to receive when he learns $x_{S_{A,1}}$ and announces z in Stage 1, receives q from the mechanism in Stage 2, and chooses $S_{A,2}$ in Stage 3. For any z, any $x_{S_{A,1}}$, and any q, let

$$S_{A,2}^*(z, q, x_{S_{A,1}}) \in \arg\max_{S_{A,2}} \nu_2(z, q, x_{S_{A,1}}, S_{A,2}).^{11}$$

Proceeding by backward induction, for any set $S_{A,1}$, we define

$$z^*(x_{S_{A,1}}) \in \arg\max_z \mathop{\mathbb{E}}_{x \leftarrow X|x_{S_{A,1}}, q \leftarrow \mathcal{D}_1} \big[\nu_2\big(z, q, x_{S_{A,1}}, S_{A,2}^*(z, q, x_{S_{A,1}})\big)\big],$$

$$\nu_1(S_{A,1}) = \mathop{\mathbb{E}}_{x \leftarrow X, q \leftarrow \mathcal{D}_1} \big[\nu_2(z^*(x_{S_{A,1}}), q, x_{S_{A,1}}, S_{A,2}^*(z, q, x_{S_{A,1}})\big],$$

so that $\nu_1(S_{A,1})$ is the agent's expected reward when he chooses set $S_{A,1}$.

Let $S_{A,1}^* \in \arg\max_S \nu_1(S)$. From Case 1, we can make the following inferences:

- For any $S_{A,1}$, the agent's choice of set $S_{A,2}^*(z^*(x_{S_{A,1}}), q, S_{A,1})$ in Stage 3 will give him enough information to report v_q such that $\|v - w\|_2^2 < \delta$.

[11] There may be multiple such sets, but we shall argue that this multiplicity does not matter for our argument.

- Given that the agent is reporting such a v_q in Stage 3, the agent chooses $S^*_{A,1}$ in Stage 1 to get enough information to learn $z = z^*(x_{S_{A,1}})$ such that $z = \sum_{q=0}^{2n} \Phi_q(v_q)$.

We now proceed to scale the reward so that, even when there is cost, the agent is always incentivized to choose $S^*_{A,1}$ in Stage 1 and is incentivized to choose $S^*_{A,2}(\cdot,\cdot,\cdot)$ in Stage 3. Choose $\kappa > 0$ such that the following conditions hold:

- For any $S_{A,1} \neq S^*_{A,1}$ and any $S_{A,2}$, we have

$$\kappa \cdot \nu_1\big(S^*_{A,1}\big) - c\big(\big|S^*_{A,1}\big| + |S_{A,2}|\big) > \kappa \cdot \nu(S_{A,1}) - c\big(|S_{A,1}| + |S_{A,2}|\big).$$

- For any z, any q, and any $x_{S_{A,1}}$, and any $S_{A,2} \neq S^*_{A,2}(z, q, S_{A,1})$,

$$\begin{aligned}&\kappa\nu_2\big(z, q, x_{S_{A,1}}, S^*_{A,2}\big(z^*(x_{S_{A,1}}), q, S_{A,1}\big)\big) - c\big(|S_{A,1}| + \big|S^*_{A,2}(z, q, S_{A,1})\big|\big)\\ &\quad > \kappa\nu_2(z, q, x_{S_{A,1}}, S_{A,2}) - c\big(|S_{A,1}| + |S_{A,2}|\big).\end{aligned}$$

Such a κ exists because $S^*_{A,2}(z, q, S_{A,1})$ and $S^*_{A,1}$ are chosen to maximize ν_1 and ν_2, respectively. Let $\tilde{R}(\cdot,\cdot) \stackrel{\text{def}}{=} \kappa \cdot R(\cdot,\cdot)$ be a scaled reward function. Let $\tilde{\nu}_1 \stackrel{\text{def}}{=} \kappa \cdot \nu_1$, $\tilde{\nu_2} \stackrel{\text{def}}{=} \kappa \cdot \nu_2$. Then, by our choice of κ, we have the following conditions:

- For any $S_{A,1} \neq S^*_{A,1}$ and any $S_{A,2}$, we have

$$\tilde{\nu_1}\big(S^*_{A,1}\big) - c\big(\big|S^*_{A,1}\big| + |S_{A,2}|\big) > \tilde{\nu}(S_{A,1}) - c\big(|S_{A,1}| + |S_{A,2}|\big).$$

- For any z, any q, and any $x_{S_{A,1}}$, and any $S_{A,2} \neq S^*_{A,2}(z^*(x_{S_{A,1}}), q, S_{A,1})$,

$$\begin{aligned}&\tilde{\nu_2}\big(z, q, x_{S_{A,1}}, S^*_{A,2}\big(z^*(x_{S_{A,1}}), q, S_{A,1}\big)\big) - c\big(|S_{A,1}| + \big|S^*_{A,2}\big(z^*(x_{S_{A,1}}), q, S_{A,1}\big)\big|\big)\\ &\quad > \tilde{\nu_2}(z, q, x_{S_{A,1}}, S_{A,2}) - c\big(|S_{A,1}| + |S_{A,2}|\big).\end{aligned}$$

Thus, by changing the reward of function of $\mathcal{M}_2$ to be $\kappa \cdot R$, we can incentivize the agent to gather enough information to be able to report v such that $\|v - w\|_2^2 < \delta$ and report z such that $|z - f(x)| < \epsilon$.

We conclude by noting that we can further increase κ to ensure individual rationality, and even if there are multiple optimal choices for $S^*_{A,1}$ and $S^*_{A,2}$, all such choices guarantee that the agent gathers enough information to report z within ϵ of $f(x)$. □

4.3 *Our fourth theorem*

Our mechanism $\mathcal{M}_2$ queries only one coordinate x_p from the input vector, and uses two rounds of interaction between the principal and the agent. This raises the question of whether there exist one-round contracts that can be used to approximately delegate continuous functions and query few inputs. We show in the following theorem that this cannot be the case as long as the reward function is concave in the agent's reported value z. We remark that the reward function $R((z, q, v_q, p), x_p)$ used in Theorem 3 is concave in the agent's reported value z, but it is not concave as a function of the agent's additional message v_q.

THEOREM 4. *There exists a continuous function $f : [-1, 1]^n \to \mathbb{R}$ such that every one-round, ϵ-incentive-compatible contract $\mathcal{M} = (\mathcal{D}, R)$ with a concave reward function must query the whole input $(x_1, \ldots, x_n)$.*

The proof of Theorem 4 is given in Appendix C.

5. DISCUSSION

Optimality and unlimited liability

In the above discussions, we have been careful to minimize the number of queries that the mechanism makes to the input (as well as the number of interaction rounds between the principal and the agent). In addition to the query cost, the principal of course bears the monetary cost of paying the agent's reward. As we mentioned in the Introduction, this monetary cost can be minimized for any of our mechanisms when the agent has unlimited liability. The principal can simply charge the agent a fixed fee equal to the agent's expected utility from participating in the mechanism. Since this fixed fee can be computed using expectations over X (rather than querying the actual input x), the principal can compute this reward at no extra cost. Since the principal minimizes her expected payment to the agent and the number of queries she makes, our contracts are optimal in the unlimited liability scenario.

In the classical setting, this type of argument is known as "selling the firm" Hölmstrom (1979) and makes the problem trivial. In this setting, the problem is still nontrivial because the principal still needs to be convinced that the agent has computed a correct approximation to $f(x)$ and, therefore, needs to query some set S of inputs so as to verify the agent's answer. Thus, even when we sell the firm and minimize the principal's monetary cost, the problem of minimizing the principal's query cost still stands.

Beyond ϵ-incentive compatibility

In this paper, we focused on the case where the principal wants to obtain a value z arbitrarily close to the true value of $f(x)$. While we believe that this is an important model for delegation of computation, it is not the only one. For instance, one may also investigate various trade-offs between the quality of the agent's answer z and the cost of obtaining this answer.

Accounting for algorithmic cost

In our results, we assumed that data collection is expensive, but once the data are available, running an algorithm on the collected data is free. An advantage of this model over other ways to measure computational complexity is that we can precisely analyze how many pieces of data a principal has to observe so as to incentivize the agent to evaluate a continuous function of the entire data set. Indeed, we showed that the principal only needs to observe a single piece of data.

An alternative approach would be to assume that the data are already available (and thus cost-free), while running algorithms on the available data has a cost that increases

with the number of operations performed. We are also interested in delegating computation to a rational agent in this model. We believe that doing so is likely to require techniques from computer science, such as computationally sound interactive proof systems (Micali 2001, Kilian 1992), where the agent gives verifiable evidence to the principal that his answer is correct. This verifiable evidence guarantees to the principal that no malicious agent (who may go out of his way to cheat her) will be able to deceive her. Such guarantees are very strong, but require very complicated protocols. We believe that by properly modeling the rationality of the agent, as we have done in this paper, we can vastly simplify these protocols.

Appendix A: Proof of Theorem 2

Theorem 2. *There exists a continuous, bounded and nonseparable function* $f : [-1, 1]^n \to \mathbb{R}$ *such that any direct computational contract* $\mathcal{M} = (\mathcal{D}, R)$ *that is incentive-compatible for* f *must query the whole input vector* $(x_1, \ldots, x_n)$.

Proof. We consider the function $f(x_1, \ldots, x_n) = x_1 x_2 \ldots x_n$. Assume that there is a direct computational contract $\mathcal{M} = (\mathcal{D}, R)$ that is incentive-compatible for f and that queries $n-1$ inputs. Let $S_{-i} = \{1, \ldots, n\} - \{i\}$ and let $x_{-i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$.

We can write the function $f(x_1, \ldots, x_n) = x_1 x_2 \ldots x_n$ as

$$x_1 x_2 \ldots x_n = \arg\max_z \sum_{i=1}^{n} \text{Prob}(\mathcal{D} = S_{-i}) R(z, x_{-i}). \tag{2}$$

Let $\phi_{-i}(x_{-i}, z) = \text{Prob}(\mathcal{D} = S_{-i}) R(x_{-i}, z)$ and let $\rho(x_1, \ldots, x_n, z) = \sum_{i=1}^{n} \phi_{-i}(x_{-i}, z)$. Then we can write $f(x_1, \ldots, x_n)$ as

$$x_1 \ldots x_n = \arg\max_z \rho(x_1, \ldots, x_n, z).$$

Now consider the set $A = \{-1, 1\}^n = \{(x_1, \ldots, x_n) : x_i \in \{1, -1\}\}$ and notice that $f(x_1, \ldots, x_n) \in \{1, -1\}$ for any vector $x \in A$. Let $A_1 = \{x \in A : f(x) = 1\}$ and $A_{-1} = \{x \in A : f(x) = -1\}$. For any vector $x \in A_1$, we must have that $1 = f(x) = \arg\max_z \rho(x, z)$, so that $\rho(x, 1) > \rho(x, -1)$. Similarly, for any $x \in A_{-1}$, we must have that $\rho(x, -1) > \rho(x, 1)$.

The above argument implies that

$$\sum_{x \in A_1} \rho(x, 1) > \sum_{x \in A_1} \rho(x, -1), \tag{3}$$

$$\sum_{x \in A_{-1}} \rho(x, -1) > \sum_{x \in A_{-1}} \rho(x, 1). \tag{4}$$

We proceed to obtain a contradiction by showing that

$$\sum_{x \in A_1} \rho(x, 1) = \sum_{x \in A_{-1}} \rho(x, 1), \tag{5}$$

$$\sum_{x \in A_1} \rho(x, -1) = \sum_{x \in A_{-1}} \rho(x, -1). \tag{6}$$

Note that for every $x \in A_1$ and index $i \in \{1, \dots, n\}$, there exists a unique $y(x, i) \in A_{-1}$ such that $y_{-i}(x, i) = x_{-i}$. Such a $y(x, i)$ can be constructed by setting $y_j(x, i) = x_j$ for all $j \neq i$ and $y_i(x, i) = -x_i$. It is clear from this construction that $x_1 \dots x_n = -y_1 \dots y_n$, and since $x_1 \dots x_n = 1$, we must have $y_1 \dots y_n = -1$. For a fixed i, it is clear that the function mapping x to $y(x, i)$ is a bijection between A_1 and A_{-1}.

To show that (5) holds, write

$$\begin{aligned}\sum_{x \in A_1} \rho(x, 1) &= \sum_{x \in A_1} \sum_{i=1}^{n} \phi_{-i}(x_{-i}, 1) = \sum_{i=1}^{n} \sum_{x \in A_1} \phi_{-i}(x_{-i}, 1) \\ &= \sum_{i=1}^{n} \sum_{x \in A_1} \phi_{-i}\big(y_{-i}(x, i), 1\big) = \sum_{i=1}^{n} \sum_{y \in A_{-1}} \phi_{-i}(y_{-i}, 1) \\ &= \sum_{y \in A_{-1}} \sum_{i=1}^{n} \phi_{-i}(y_{-i}, 1) = \sum_{y \in A_{-1}} \rho(y, 1).\end{aligned}$$

Analogously, we can show that (6) holds. Now note that from (3), (4), (5), and (6) we can derive the contradiction

$$\sum_{x \in A_1} \rho(x, 1) > \sum_{x \in A_1} \rho(x, -1) = \sum_{x \in A_{-1}} \rho(x, -1) > \sum_{x \in A_{-1}} \rho(x, 1) = \sum_{x \in A_1} \rho(x, 1).$$

This contradiction arose from assuming that $x_1 \dots x_n$ could be written in the form of (2). Thus, we conclude that $x_1 \dots x_n$ cannot be delegated with an $(n-1)$-query direct revelation contract. □

To prove this theorem we have used the fact that the contract is strictly incentive-compatible. The proof would still work with weakly incentive-compatible contracts.[12]

Appendix B: Proof of Lemma 1

Lemma 1. *Let v and w be real numbers in $[0, 1]$, and let V and W be random variables over $\{0, 1\}$ such that $\Pr(V = 1) = v$ and $\Pr(W = 1) = w$. Then*

$$\mathop{\mathbb{E}}_{\omega \leftarrow V}\big[\mathrm{BSR}(V, \omega) - \mathrm{BSR}(W, \omega)\big] = 2(v - w)^2.$$

[12] We illustrate this informally. Suppose that the distribution of inputs X is such that $x \in A_1$ with probability $1 - \epsilon$ and $x \in A_{-1}$ with probability ϵ. Suppose, furthermore, that the contract is weakly incentive-compatible so that for any $x \in A_1$ and any $\tilde{x} \in A_{-1}$, we have

$$\mathop{\mathbb{E}}_{S \leftarrow \mathcal{D}}\big[R\big(f(x), x_S\big)\big] \geq \mathop{\mathbb{E}}_{S \leftarrow \mathcal{D}}\big[R\big(f(\tilde{x}), x_S\big)\big],$$

$$\mathop{\mathbb{E}}_{S \leftarrow \mathcal{D}}\big[R\big(f(\tilde{x}), \tilde{x}_S\big)\big] \geq \mathop{\mathbb{E}}_{S \leftarrow \mathcal{D}}\big[R\big(f(x), \tilde{x}_S\big)\big].$$

If any one of these equalities holds strictly, we derive a contradiction just as in our proof above. If both hold as equalities, then the contract cannot be weakly incentive-compatible because the agent knows that the value of f is equal to $f(x) = 1$ with extremely high probability and, therefore, will maximize his expected payoff by always reporting the value 1 without querying any inputs, and, thus, obtaining the same reward with less cost.

PROOF. Note that we have $\|V\|_2^2 = v^2 + (1-v)^2$ and $\|W\|_2^2 = w^2 + (1-w)^2$. We have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\omega \leftarrow V} \mathrm{BSR}(V, \omega) \\
&\quad = \Pr(\omega = 1)\big(2\Pr(V = 1) - \|V\|_2^2 + 1\big) + \Pr(\omega = 0)\big(2\Pr(V = 0) - \|V\|_2^2 + 1\big) \\
&\quad = 2v^2 + 2(1-v)^2 - v^2 - (1-v)^2 + 1 = v^2 + (1-v)^2 + 1 \\
&\quad = 2v^2 - 2v + 2.
\end{aligned}$$

We also have

$$\begin{aligned}
\mathop{\mathbb{E}}_{\omega \leftarrow V} \mathrm{BSR}(W, \omega) &= \Pr(\omega = 1)\big(2\Pr(W = 1) - \|W\|_2^2 + 1\big) \\
&\quad + \Pr(\omega = 0)\big(2\Pr(W = 0) - \|W\|_2^2 + 1\big) \\
&= 2vw + 2(1-v)(1-w) - w^2 - (1-w)^2 + 1 \\
&= 4vw + 2 - 2v - 2w^2.
\end{aligned}$$

Taking differences, we have

$$\begin{aligned}
\mathop{\mathbb{E}}_{\omega \leftarrow V}\big[\mathrm{BSR}(V, \omega) - \mathrm{BSR}(W, \omega)\big] &= 2\big(v^2 + w^2\big) - 2v - 4vw + 2v \\
&= 2v^2 + 2w^2 - 4wv \\
&= 2(v-w)^2.
\end{aligned}$$

□

APPENDIX C: PROOF OF THEOREM 4

The following lemma will be useful for the proof of Theorem 4.

LEMMA 2. *Let $\phi_1, \ldots, \phi_n : \mathbb{R} \to \mathbb{R}$ be continuous concave functions, and let $\phi : \mathbb{R} \to \mathbb{R}$ be a function such that $\phi(z) = \sum_{i=1}^n \phi_i(z)$. Let $z_i^* = \arg\max_z \phi_i(z)$ and assume that there exists an interval $[a, b]$ such that for all i, $z_i^* \in [a, b]$. Then*

$$\arg\max_z \phi(z) \in [a, b].$$

PROOF. Since each ϕ_i is continuous and concave, it is superdifferentiable everywhere.[13] Furthermore, since each ϕ_i is maximized in the interval $[a, b]$ for each i, the following supergradients exist:

- A supergradient $\alpha_i \geq 0$ such that $\phi_i(z) - \phi_i(a) \leq \alpha_i(z - a)$ for every $z \in \mathbb{R}$.
- A supergradient $\beta_i \leq 0$ such that $\phi_i(z) - \phi_i(b) \leq \beta_i(z - b)$ for every $z \in \mathbb{R}$.

[13] A function $f : \mathbb{R} \to \mathbb{R}$ is superdifferentiable at x if there exists a real number c such that for every $z \in \mathbb{R}$, we have $f(z) - f(x) \leq c(z - x)$.

Adding up over all i, we conclude that there exist supergradients $\alpha = \sum_{i=1}^{n} \alpha_i \geq 0$ and $\beta = \sum_{i=1}^{n} \beta_i \leq 0$ for ϕ such that $\phi(z) - \phi(a) \leq \alpha(z-a)$ and $\phi(z) - \phi(b) \leq \beta(z-b)$ for every $z \in \mathbb{R}$. Since ϕ is concave, this implies that there exists a point $z^* \in [a,b]$ such that 0 is a supergradient of ϕ at z^*. That is, $\phi(z) - \phi(z^*) \leq 0(z-z^*)$ for every $z \in \mathbb{R}$. This implies that

$$\phi(z) \leq \phi\big(z^*\big)$$

for every $z \in \mathbb{R}$. Therefore, the maximum of ϕ is in the interval $[a,b]$. □

THEOREM 4. *There exists a continuous function $f : [-1,1]^n \to \mathbb{R}$ such that every one-round, ϵ-incentive-compatible contract $\mathcal{M} = (\mathcal{D}, R)$ with a concave reward function must query the whole input $(x_1, \ldots, x_n)$.*

PROOF. As in the proof of Theorem 2, we consider the function $f(x_1, \ldots, x_n) = x_1 x_2 \ldots x_n$. Let $\epsilon < 1$ and assume that an $(n-1)$-query, ϵ-incentive-compatible contract $\mathcal{M} = (\mathcal{D}, R)$ exists for delegating f. We can then write the agent's expected reward from sending a message m as

$$\sum_{i=1}^{n} \text{Prob}(\mathcal{D} = S_{-i}) R(m, x_{-i}) = \sum_{i=1}^{n} \phi_{-i}(m, x_{-i}),$$

where $\phi_{-i}(m, x_{-i}) = \text{Prob}(\mathcal{D} = S_{-i}) R(m, x_{-i})$. Note that each $\phi_{-i}(m, x_{-i})$ is concave. Therefore,

$$\rho(m, x) = \sum_{i=1}^{n} \phi_{-i}(m, x_{-i})$$

will also be concave.

Define the set $A = \{-1,1\}^n = \{(x_1, \ldots, x_n) : x_i \in \{1,-1\}\}$ and notice that $f(x_1, \ldots, x_n) \in \{1,-1\}$ for any vector $x \in A$. Let $A_1 = \{x \in A : f(x) = 1\}$ and $A_{-1} = \{x \in A : f(x) = -1\}$. Recall that for every $x \in A_1$ and every index $i \in \{1, \ldots, n\}$, there exists a unique $y(x,i) \in A_{-1}$ such that $x_{-i} = y_{-i}(x,i)$. The key to proving Theorem 4 is using this bijection to derive the equation for any possible message m:

$$\sum_{x \in A_1} \rho(m, x) = \sum_{y \in A_{-1}} \rho(m, y). \tag{7}$$

This equation holds because

$$\begin{aligned}
\sum_{x \in A_1} \rho(m, x) &= \sum_{x \in A_1} \sum_{i=1}^{n} \phi_{-i}(m, x_{-i}) = \sum_{i=1}^{n} \sum_{x \in A_1} \phi_{-1}(m, x_{-i}) \\
&= \sum_{i=1}^{n} \sum_{x \in A_1} \phi_{-i}\big(m, y_{-i}(x,i)\big) = \sum_{i=1}^{n} \sum_{y \in A_{-1}} \phi_{-i}(m, y_{-i}) \\
&= \sum_{y \in A_{-1}} \sum_{i=1}^{n} \phi_{-i}(m, y_{-i}) = \sum_{y \in A_{-1}} \rho(m, y).
\end{aligned}$$

We now show that (7) leads to a contradiction. Since $\rho(m,x)$ is a continuous and concave function, there exists a continuous function $m(x)$ such that $m(x) = \arg\max_m \rho(m,x)$. Since the mechanism is ϵ-incentive-compatible, $m(x) \in [f(x) - \epsilon, f(x) + \epsilon]$. In particular, for any $x \in A_1$, we must have $m(x) \in [1-\epsilon, 1+\epsilon]$, and for any $y \in A_{-1}$, we must have $m(y) \in [-1-\epsilon, -1+\epsilon]$.

The sum $\sum_{x\in A_1} \rho(m,x)$ is itself a concave function of m. Let $m_1^* = \arg\max_m \sum_{x\in A_1} \rho(m,x)$. Since each $m(x) = \arg\max_m \rho(m,x)$ belongs to the interval $(1-\epsilon, 1+\epsilon)$, Lemma 2 tells us that $m_1^* \in [1-\epsilon, 1+\epsilon]$. Analogously, the same lemma tells us that $m_{-1}^* = \arg\max_m \sum_{y\in A_1} \rho(m,y)$ must belong in the interval $[-1-\epsilon, -1+\epsilon]$.

But (7) tells us that

$$\sum_{x\in A_1} \rho\big(m_1^*, x\big) = \sum_{y\in A_{-1}} \rho\big(m_1^*, y\big),$$

$$\sum_{y\in A_{-1}} \rho\big(m_{-1}^*, y\big) = \sum_{x\in A_1} \rho\big(m_{-1}^*, x\big).$$

The strict concavity of ρ implies that $\sum_{y\in A_{-1}} \rho(m_{-1}^*, y) > \sum_{y\in A_{-1}} \rho(m_1^*, y)$ and $\sum_{x\in A_1} \rho(m_1^*, x) > \sum_{x\in A_1} \rho(m_{-1}^*, x)$. Putting these equations together, we derive the contradiction

$$\begin{aligned}
\sum_{y\in A_{-1}} \rho\big(m_{-1}^*, y\big) &= \sum_{x\in A_1} \rho\big(m_{-1}^*, x\big) \\
&< \sum_{x\in A_1} \rho\big(m_1^*, x\big) \\
&= \sum_{y\in A_{-1}} \rho\big(m_1^*, y\big) \\
&< \sum_{y\in A_{-1}} \rho\big(m_{-1}^*, y\big).
\end{aligned}$$

From this contradiction, we conclude that an $(n-1)$-query, ϵ-incentive-compatible contract for $f(x_1, \ldots, x_n) = x_1 \ldots x_n$ does not exist. □

As a final remark, note that we do not need f to be symmetric for Theorem 4 to hold. The proof tells us that there exists a symmetric function f such that any one-round ϵ-incentive-compatible contract for f requires the principal to query all n coordinates of x. However, we can construct a nonsymmetric perturbation $\tilde{f}$ of f for which Theorem 4 also holds. Let $g(x)$ be a nonsymmetric continuous function such that $0 \le g(x) \le 1$. Let $\tilde{f}(x) = f(x) + \frac{\epsilon}{2} \cdot g(x)$. Then we have that $\tilde{f}$ is not symmetric and that $|f(x) - \tilde{f}(x)| \le \frac{\epsilon}{2}$. Even though $\tilde{f}$ is not symmetric, there does not exist an $\frac{\epsilon}{2}$-incentive-compatible one-round computational contract $(\tilde{\mathcal{D}}, \tilde{\mathcal{R}})$ for $\tilde{f}$ that queries $k < n$ coordinates of x. If such a contract existed, then it would also be an ϵ-incentive-compatible contract for f. This is because the contract would incentivize the agent to reveal some value $\tilde{m}(x)$ such that $|\tilde{m}(x) - \tilde{f}(x)| \le \frac{\epsilon}{2}$. The revealed value $\tilde{m}(x)$ would also satisfy $|\tilde{m}(x) - f(x)| \le \epsilon$.

We conclude that there is a very general class of functions for which no one-round mechanisms that make $k < n$ queries can be ϵ-incentive-compatible.

References

Al-Najjar, Nabil I., Luca Anderlini, and Leonardo Felli (2006), "Undescribable events." *Review of Economic Studies*, 73, 849–868. [558]

Anderlini, Luca and Leonardo Felli (1994), "Incomplete written contracts: Undescribable states of nature." *Quarterly Journal of Economics*, 109, 1085–1124. [558]

Azar, Pablo D. and Micali Silvio (2012), "Rational proofs." In *Proceedings of the 44th Annual ACM symposium on Theory of Computing (STOC'12)*, 1017–1028. ACM, New York. [558]

Carroll, Gabriel (2017), "Robust incentives for information acquisition." Unpublished paper, Stanford University. [558]

Clemen, Robert T. (2002), "Incentive contrats and strictly proper scoring rules." *Test*, 11, 167–189. [557, 562]

CERN Computing. Available at http://home.web.cern.ch/about/computing. [554]

Demski, Joel S. and David E. M. Sappington (1987), "Delegated expertise." *Journal of Accounting Research*, 25, 68–89. [557]

Gneiting, Tilmann and Adrain E. Raftery (2007), "Strictly proper scoring rules, prediction, and estimation." *Journal of the American Statistical Association*, 102, 359–378. [561]

Guo, Siyao, Pavel Hubacek, Alon Rosen, and Margarita Vald (2015), "Rational sum-checks." In *Theory of Cryptography: 13th International Conference, TCC 2016-a, tel Aviv, Israel, January 10–13, 2016, Proceedings, Part 2* (Eyal Kushilevitz and Tal Malkin, eds.), 319–351, Springer, Berlin. [558]

Hölmstrom, Bengt (1979), "Moral hazard and observability." *Bell Journal of Economics*, 10, 74–91. [556, 571]

Kilian, Joe (1992), "A note on efficient zero-knowledge proofs and arguments." In *ACM Symposium on Theory of Computing*, 723–732, ACM, New York, NY. [572]

Kolmogorov, Andrey N. (1963), "On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition." *American Mathematical Society Translations: Series 2*, 28, 55–59. [565]

Lambert, Nicolas S. (2013), "Elicitation and evaluation of statistical forecasts." Unpublished paper, Graduate School of Business, Stanford University. [557, 562]

Malcolmson, James M. (2009), "Principal and expert agent." *B.E. Journal of Theoretical Economics*, 9, Article ID 17. [558]

MacLeod, William B. (2000), "Complexity and contract." *Revue d'conomie Industrielle*, 92, 149–178. [558]

Micali, Silvio (2001), "Computationally sound proofs." *SIAM Journal on Computing*, 30, 1253–1298. [572]

Osband, Kent (1989), "Optimal forecasting incentives." *The Journal of Political Economy*, 97, 1091–1112. [557, 562]

Zermeño Vallés, Luis G. (2012), *Essays on the Principal-Expert Problem*. Ph.D. thesis, Massachusetts Institute of Technology, Dept. of Economics. [558]

Co-editor George J. Mailath handled this manuscript.

Manuscript received 26 March, 2014; final version accepted 19 August, 2017; available online 20 September, 2017.