Prediction Games
(Extended Abstract)

Amir Ronen[1,⋆] and Lyron Wahrmann[2,⋆⋆]

[1] Faculty of Industrial Engineering & Management, Technion,
Technion City, Haifa 32000, Israel
amirr@ie.technion.ac.il

[2] Department of Mathematics, Technion,
Technion City, Haifa 32000, Israel
lyron@tx.technion.ac.il

Abstract. We introduce a new class of mechanism design problems called *prediction games*. There are n self interested agents. Each agent i has a *private* input x_i and a cost of accessing it. Given are a function $f(x_1, \ldots, x_n)$ which predicts whether a certain event will occur and an independent distribution on the agents' inputs. Agents can be rewarded in order to motivate them to access their inputs. The goal is to design a mechanism (protocol) which in equilibrium predicts $f(.)$ and pays in expectation as little as possible.

We investigate both, exact and approximate versions of the problem and provide several upper and lower bounds. In particular, we construct an optimal mechanism for every anonymous function and show that any function can be approximated with a relatively small expected payment.

1 Introduction

1.1 Motivation

Predictions of future events play an important role in our everyday life. Individuals want for example to know whether it will rain tomorrow, and who will win the next elections. Companies may be interested whether a marketing campaign will succeed. Governments want to predict the usefulness of public projects, etc.

A powerful method of making accurate predictions is the usage of input provided by multiple agents. Often, such predictions are more accurate than those performed by a single agent, even an expert (see e.g [2, 3, 15, 16]). Currently, multi agent predictions are mainly carried out by public opinion polls, phone surveys, and in a more limited scale by future markets. Moreover, many organizations (e.g. governments, institutions, companies) base predictions on input of external experts which do not necessarily share the interest of the organization. A major difficulty in making such predictions is that the participating agents have no incentive to spend the time and effort necessary for providing their inputs (answering long questionnaires, perform costly checks, etc.).

⋆ Supported in part by grant number $53/03 - 10.5$ from the Israel Science Foundation, a grant from Matilda Barnett Revocable Trust, and an award from the Jacob Ullman Memorial Academic Lectureship.

⋆⋆ Supported in part by grant number $53/03 - 10.5$ from the Israel Science Foundation.

X. Deng and Y. Ye (Eds.): WINE 2005, LNCS 3828, pp. 129–140, 2005.

Naturally, mechanisms which ask costly inputs from the agents need also to reward them. As we shall see, such rewards may need to be very high in order to ensure that the agents are motivated and developing low payment prediction mechanisms is a non trivial task. In order to demonstrate this, let us consider the following toy example.

Example (consensus). A company develops a new exciting product. It would like to predict whether the product will be ready before the holiday season. Five managers are involved in the development process. Each manager has information which indicates the chances that the company's deadline will be met. In view of the company's past experience the product is likely to be ready on time if and only if all managers predict so. This is an example of a *consensus* function. The event that the company wants to predict is whether the product will be ready on time and the inputs are the managers' indications. Assume that each manager has a positive indication with probability $1/2$ and that the time required for each manager to issue an estimation costs her $\$1,000$.

Lets examine a few intuitive methods by which the company can try to forecast its success. One option is to ask every manager to predict whether the deadline will be met and eventually reward only the managers that were correct. Consider one of the managers. The probability that another manager will have a negative indication is $15/16$. Thus, even when the manager's input is positive, the probability of meeting the deadline is only $1/16$. Therefore, all managers are better off reporting a negative estimation regardless of their actual information.

A better option is to ask each manager to report her *own* indication and reward *all* of them if and only if their **joint** prediction is correct. Consider one of the managers, say Alice. If another manager has a negative indication then Alice's input does not matter to the prediction. Thus, the probability that Alice's input matters is $1/16$. In addition, Alice can guess her input with probability $1/2$. Intuitively, the lower the chances that Alice's input is relevant, the less motivated will she be to invest the costly effort of performing her evaluation. Indeed, we shall show later that unless a manager is rewarded by at least $\frac{\$1,000}{1/2 \cdot 1/16} = \$32,000$ she will be better off guessing her input. In such a case the prediction of the company is arbitrary. Therefore, the intuitive mechanism which simultaneously asks the participants to compute and report their inputs must pay a sum of at least $\$160,000$, even though the actual cost of the managers is only $\$5,000$! We will show later that it is possible to design a mechanism which predicts the consensus function with an overwhelmingly smaller expected payment.

1.2 Our Contribution

We introduce a new class of mechanism design problems called *prediction games*. There are n self interested agents. Each agent i holds a *private* input bit x_i. The cost for each agent of accessing its input is c. Given is a function $f(x_1, \ldots, x_n) \to \{0,1\}$ which predicts whether a certain event will occur. The mechanism can pay the agents in order to motivate them to access their inputs. Given an IID (Independent Identical Distribution) on the agents' inputs, the goal is to design a mechanism which in equilibrium predicts $f(.)$ and pays in expectation as little as possible.

We show that it is possible to focus on a very specific class of mechanisms. These mechanisms approach the agents serially (each agent at most once), do not reveal *any* information to the agents other than the fact that they are required to compute their input, stop when $f(.)$ is determined, and pay according to a specific scheme. In equilibrium, each agent approached computes its input and discloses it to the mechanism. We call such mechanisms *canonical*.

Perhaps the most important prediction functions are anonymous (agent symmetric) functions. These functions include majority, consensus, percentile, etc. We show that the **optimal** mechanisms for anonymous functions are canonical mechanisms which approach the agents in random order.

We show interesting connections between the expected payment of prediction mechanisms and the analysis of the influence of random variables. Using results from influence analysis, we construct upper bounds on the expected payment for both general and anonymous functions. These results also bound the sum of utilities required for computing a function in the model of [6, 7, 11].

The necessary expected payment for the exact prediction of many functions is very high. Therefore, we study approximate prediction mechanisms. These are mechanisms that allow a small probability of error. We show that **every** function can be approximated with a relatively low expected payment.

Finally, several important prediction functions are analyzed in the paper.

Our setup is basic and there is a lot of room for extensions. Yet, we believe that the insights gained in this paper apply to many real life prediction scenarios.

1.3 Related Work

Most of the vast literature on voting and group decision making assumes that the agents have free access to their inputs. Several recent papers [6, 7, 11] analyze situations of decision making with costly inputs. (A somewhat different setup was studied in [4, 5].) In these papers each agent has a utility for every possible decision and payments are not used by the mechanism. [6, 7] mainly focus on anonymous functions and [11] is dedicated to majority rules. The main concern of these papers is to characterize the functions which can be implemented in equilibrium.

Our model is similar to the model used in [6, 7, 11] but there are fundamental differences. Firstly, we allow the mechanism to *pay* the agents. Secondly, we assume that the agents are indifferent about the prediction of the mechanism. The utility of an agent is solely determined by its payment and the cost of accessing its input. Finally, correctness of the prediction can be *verified* by the mechanism.

Therefore, in our model every function can be implemented, and our main concern is how to minimize the expected payment of the mechanism.

Markets, and future markets in particular are known to be good predictors in certain situations. Recently, several artificial markets for forecasting have been established. Among the examples are the IOWA electronic markets (http://www.biz.uiowa.edu/iem/), Hollywood Stock Exchange (http://www.hsx.com/), and the Foresight Exchange site (http://www.ideosphere.com/). Such markets were studied in several papers (e.g. [2, 3]). A pioneering paper [1] provides a theoretical analysis of the power of future markets. The paper shows that many functions can be predicted (under strong assumptions such as myopic behavior of the agents) but others, such as parity, cannot.

More empirical literature exists within the forecasting community. A comparison between the game theoretic approach and other methods used in conflict forecasting can be found in articles published in the International Journal of Forecasting (e.g. [14],[15]).

Our model is based on mechanism design, a subfield of game theory and micro economics that studies the design of protocols for self interested parties. An introduction to this field can be found in many textbooks (e.g. [13–chapter 23]).

We show connections between the payment that must be made to an agent and its influence on the prediction function (the probability that its input is necessary for making the prediction). Since the seminal work of [8] influence analysis evolved significantly. Its focus is usually on bounds on the aggregate and the maximum influence. While we are usually interested in the harmonic mean, some of these results are still very helpful to us. A recent survey on influence literature can be found in [10].

Note: Due to space constraints large parts of the article were omitted from the proceedings. The full version can be found at http://iew3.technion.ac.il/ amirr/.

2 Preliminaries

2.1 The Model

In this section we introduce our model and notation. We have tried as much as possible to adopt the standard approach of mechanism design theory.

There are n self interested agents. Each agent i holds a *private* input bit x_i. The input $x = (x_1, \ldots, x_n)$ is taken from a prior distribution ϕ which is common knowledge. Each agent can access its input x_i but this access is costly. For simplicity of the presentation we assume that all agents have the same cost c. Our results hold when each agent has a different (but known) cost as well. An agent can also *guess* its input. We assume that guessing is free so the cost of an agent is either c in case it computed its input or zero. We allow the mechanism to pay the agents (for otherwise none will access its input) but not to fine them. The *utility* of each agent is the difference between its payment and cost. Each agent selfishly tries to maximize its expected utility.

We are given a boolean *prediction function* $f : \{0,1\}^n \rightarrow \{0,1\}$ which given the inputs of the agents predicts whether a certain event will occur.

A *prediction mechanism* is a communication protocol and a payment function. After communicating with the agents the mechanism announces whether the event will occur. It pays the agents *after* it is known whether the event occurred, i.e. after $f(x)$ is revealed. The mechanism is known to the participating agents.

A *strategy* for an agent is a function from its input to its behavior during the execution of the mechanism. A set of strategies $s = (s_1(x_1), \ldots, s_n(x_n))$ is a *Bayesian equilibrium* if each agent cannot improve its expected utility by unilaterally deviating from its strategy. An agent can always obtain a non negative utility by guessing its input. Therefore, any Bayesian equilibrium also guarantees an expected utility which is non negative (individual rationality).

Definition 1. (Implementation) *An* implementation *of a boolean function $g(x)$ is a pair (m, s) where m is a prediction mechanism and $s = (s_1(x_1), \ldots, s_n(x_n))$ is a*

Bayesian equilibrium in this mechanism, such that for every input x, the mechanism outputs $g(x)$. An implementation is called exact *if $g(x) = f(x)$.*

Note that the function $g(.)$ which the mechanism implements needs not necessarily be equal to $f(.)$. This will play an important role later when we study approximation mechanisms. It is not difficult to see that implementation in dominant strategies is impossible.

Consider the example of Section 1.1. The input of each manager is whether she has a positive indication that the deadline will be met. This input is uniformly distributed with $q = 1/2$. The prediction function $f(.)$ is consensus. The access cost c of each manager is $\$1,000$. A possible mechanism m would be to simultaneously ask each manager to evaluate her input and to reward it by $\$32,000$ iff the mechanism's prediction is correct. A possible equilibrium s in this mechanism is the truthful equilibrium: each agent submits a sincere estimation to the mechanism. In this equilibrium, each manager has an expected utility of $\$31,000$. (m, s) is an implementation of the consensus function.

Definition 2. (Expected payment) *Let (m, s) be an implementation. The* expected payment *$B(m)$ of the implementation is defined as the expected total payment of the mechanism when the agents follow their equilibrium strategies and the input is drawn from the underlying distribution ϕ. That is, $B(m) = \mathbf{E}[\sum_i v_i(s(x))]$ where v_i denotes the payment of each agent i.*

Similarly, the *accuracy* of the mechanism is defined as the probability that $g(x) \neq f(x)$. The analysis of influence of random variables plays an important role in our work. Most of the literature on this topic assumes that the input bits are drawn from an IID. We therefore adopt this assumption in our paper. We denote by ϕ_q the product distribution obtained when each input is independently set to 1 with probability q. We conjecture that our results also hold when there exists a $\Delta > 0$ such that each agent has its own probability $\Delta < q_i < 1 - \Delta$. However, this conjecture requires generalizations of basic results in the analysis of the influence of random variables for this setup. We assume w.l.o.g. that $0 < q < 1$, otherwise the agent's input is known to the mechanism and thus the agent is redundant. We let $q_M = \max(q, 1-q)$ denote the maximal probability in which an agent can guess its input successfully.

Notation: Let x be an n-tuple. We denote by x_{-i} the $(n-1)$-tuple $(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$.

2.2 Influence of Random Variables

Definition 3. (Influence) *Agent i is called* pivotal *for x_{-i} if $f(0, x_{-i}) \neq f(1, x_{-i})$. The* influence *$I_i$ of agent i is the probability that i is pivotal when x is chosen randomly according to ϕ_q.*

In other words, influence measures the extent of 'necessity' of agent i for making the prediction. We denote by $I_{max} = \max_i(I_i)$ the maximal influence of any agent.

Lemma 1. (Talagrand's Inequality) [9] *Let p denote the probability that $f(x) = 1$ when x is chosen according to ϕ_q. There exists a universal constant κ such that*

$$\sum_{i=1}^{n} I_i / \log(1/I_i) \geq \kappa p(1-p)q(1-q). \tag{1}$$

Lemma 2. (Maximum influence) *Let p denote the probability that $f(x) = 1$ when x is chosen according to ϕ_q. Then*

$$I_{max} = \Omega\left(p(1-p)q(1-q)\frac{\log n}{n}\right). \tag{2}$$

We often use the following well known mean inequality which states that the harmonic mean is always dominated by the arithmetic one.

Lemma 3. (Harmonic vs Arithmetic mean) *Let $a_1, \ldots, a_n$ be positive numbers. Then*

$$\frac{n}{\sum_i 1/a_i} \le \frac{1}{n}\sum_i a_i. \tag{3}$$

3 Canonical Prediction Mechanisms

In this section we shall consider exact implementations and show that it is possible to focus on a small class of mechanisms which we call canonical. This will provide us tools for both, the construction of low payment mechanisms and for proving lower bounds on the expected payment.

We start with a standard argument in mechanism design stating that it is possible to focus on mechanisms which have a very simple communication structure.

Definition 4. (Truthful implementation) *A* truthful implementation *is an implementation with the following properties:*

- *The mechanism communicates with each agent at most once asking it to report its input. It can also pass additional information to the agent. The agent then replies with either* 0 *or* 1. *We call such a mechanism* revelation *mechanism.*
- *In equilibrium, each agent approached computes its input and reveals it to the mechanism. We call such an equilibrium* truthful.

We say that two implementations are *equivalent* if for every input vector x they announce the same predictions and hand the same payments to the agents.

Proposition 1. (revelation principle for prediction mechanisms) *For every implementation there exists an equivalent truthful implementation.*

Note that there exist many truthful mechanisms. In particular, the mechanism can choose what information to pass to an agent, the payment scheme, and the order of addressing the agents.

Consider an agent which receives some information from the mechanism. The agent can compute the likelihood of every input vector of the other agents given that the mechanism passed it this information. In other words, any information which the mechanism passes to an agent defines a probability distribution on the input of the other agents. For example, consider a mechanism for consensus in which each agent knows all the declarations of the agents that were approached before it. The first agent approached has an influence of $1/2^{n-1}$. However, if all the agents except one were already approached, this agent knows that its declaration will determine the prediction. In other words, the conditional influence of this agent is 1.

Definition 5. (Conditional influence) *We denote by $I_i(\varphi_{-i})$ the* conditional influence *of agent i. That is, the probability that agent i is pivotal given a distribution φ_{-i} on the others' input (not necessarily the original distribution).*

Remark. We assume w.l.o.g. that every approached agent has a strictly positive conditional influence. Otherwise the mechanism does not need to approach the agent.

Before proceeding, let us recall a few notations that we shall use extensively. $q = \mathbf{Pr}\big[x_i = 1\big]$ for each i. We denote by $q_M = \max(q, (1-q))$ the maximal probability by which each agent can guess its input and by ϕ_q the product distribution on all the agents. The constant c denotes the cost that each agent must incur in order to access its input. We let v_i denote the expected equilibrium payment of each agent i.

Theorem 1. (Payment characterization for exact implementation) *Let m be a revelation mechanism that implements f whenever the agents are truthful. Suppose that m passes information to agent i which defines a probability distribution φ_{-i} on the inputs of the other agents. Suppose that the other agents are truthful. Let $v_0 \geq 0$ denote the expected payment of agent i in case of a wrong prediction. Truth telling is a best response for agent i iff its expected payment in case of a true prediction is at least $v_i \geq v_0 + \frac{c}{(1-q_M)\cdot I_i(\varphi_{-i})}$.*

Proof : The only inputs that the mechanism has are the agents' declarations and the actual outcome (i.e. whether the event occurred). Agent i needs to decide whether to guess its input or compute it and reveal it to the mechanism. (Since the others are truthful, the best response strategy cannot be to compute the input and then misreport it.) We need the following claim:

Claim. Under the conditions of Theorem 1:

1. The mechanism cannot distinguish between the case where agent i guesses its input correctly and a the case where it computes it.
2. If the agent is not pivotal, the mechanism cannot distinguish between the case where agent i guesses its input according to ϕ_q and the case where it computes it.

Proof : The first point is obvious as the inputs of the mechanism in both cases are identical. In the second point we use the fact that the input of the other agents is independent of agent i. Since the agent is not pivotal, the actual outcome $f(x)$ is also independent of its input. Therefore, the distribution of the input of the mechanism is identical in both cases. □

Thus, in equilibrium, the only situation in which the expected payment of the agent is different from its payment when it is truthful, is when it guesses wrong and is pivotal. In equilibrium, this happens if and only if the **prediction** of the mechanism is wrong. Recall that the expected payment of the agent in this case is v_0 and its expected payment when the prediction is correct is v_i. Recall also that the agent can guess its correct input with probability q_M. The following matrix summarizes the utility of guessing.

	Pivotal	Not Pivotal
Correct guess	$q_M \cdot I_i(\varphi_{-i})v_i$	$q_M(1 - I_i(\varphi_{-i}))v_i$
Wrong guess	$(1-q_M)I_i(\varphi_{-i})v_0$	$(1-q_M)(1 - I_i(\varphi_{-i}))v_i$

Thus, the agent's strategic considerations are expressed in the following payments matrix.

Strategy/Prediction	Correct	Wrong
Compute	$v_i - c$	$v_0 - c$
Guess	$[q_M I_i(\varphi_{-i}) + (1 - I_i(\varphi_{-i}))]\, v_i$	$(1 - q_M) I_i(\varphi_{-i}) v_0$

This matrix shows the utility of the agent according to the correctness of the prediction of the mechanism. For example, if the agent chose to compute its input and the prediction was correct its utility is $v_i - c$, since it invested c in the computation and was rewarded v_i for the correct prediction. Note that in a truthful equilibrium, it is impossible that the agent computed its input but the prediction was wrong.

Thus, a necessary and sufficient condition for an agent to compute its input is that the following inequality is satisfied,

$$v_i - c \geq (1 - q_M) I_i(\varphi_{-i}) v_0 + [I_i(\varphi_{-i}) q_M + (1 - I_i(\varphi_{-i}))] v_i \tag{4}$$

We call this inequality **Incentive Elicitation Condition (IEC)**.

The bound $v_0 + \frac{c}{(1-q_M)\cdot I_i(\varphi_{-i})}$ is then obtained by a simple calculation.

This completes the proof of Theorem 1. □

Corollary 1. *Under the conditions of Theorem 1. For every exact truthful implementation m there exists another exact truthful implementation $\tilde{m}$ with an expected payment $B(\tilde{m}) \leq B(m)$ which rewards the agents an amount of $\frac{c}{(1-q_M)\cdot I_i(\varphi_{-i})}$ iff $f(.)$ is predicted correctly.*

We call the payment scheme described in Corollary 1 *canonical*. The corollary states that w.l.o.g. we can focus on such schemes. Two important issues which the mechanism designer needs to address are what additional information to transfer to each agent and when to stop the computation.

Definition 6. (Canonical mechanisms) *A* canonical mechanism *is a truthful implementation with the following properties:*

- *The mechanism approaches the agents serially and reveals no additional information to them.*
- *The mechanism never performs redundant computations.*
- *The payment scheme is canonical. The payment v_i offered to agent i in case of a correct prediction equals $\frac{c}{(1-q_M)\cdot \hat{I}_i}$ where $\hat{I}_i$ denotes the influence of agent i conditional on the fact that it is approached.*

Theorem 2. (Optimality of a canonical mechanism) *For every function $f(.)$ there exists a canonical implementation of it with an optimal expected payment among all the exact implementations of $f(.)$.*

The theorem's proof and some other results of this section can be found in the full version of this paper.

From now on we can therefore limit ourselves to canonical mechanisms and the only decision we need to take is the *order* in which the agents are approached.

4 Exact Predictions

In this section we consider exact implementations. These are implementations which in equilibrium always issue a correct prediction.

4.1 Budget Estimations

This section is omitted due to lack of space.

4.2 Optimal Mechanisms for Anonymous Functions

This subsection characterizes the **optimal** mechanism for anonymous functions. Theorem 2 implies that the only decision left when designing a mechanism is the order in which the agents are approached. We will show that for anonymous functions *random* order is best.

Definition 7. (Anonymous functions) *Anonymous functions are functions of which the value depends only on the sum of inputs.*

In other words, anonymous functions are invariant to input permutations. These functions include majority, percentile, consensus, etc.

Definition 8. (Equal opportunity mechanism for anonymous functions) *Let $f(.)$ be an anonymous function. An* equal opportunity *mechanism for $f(.)$ is a canonical mechanism which approaches the agents according to a* random *order.*

The equal opportunity mechanism can be computed in polynomial time provided that it is possible to decide in polynomial time whether $f(.)$ has already been determined.

Theorem 3. *If $f(.)$ is anonymous, the equal opportunity mechanism is an optimal implementation of it.*

From the above theorem we can get precise lower bounds on the expected payment of various anonymous functions. Our bounds imply bounds on the sum of the utilities required for computing a function in the model of [6, 7, 11]. We now analyze the optimal mechanisms for two important functions, majority and consensus.

Theorem 4. (Lower bound for majority) *An exact prediction of majority requires an expected payment of at least $\Omega(cn^{3/2})$. The payment increases exponentially with $|q - 1/2|$.*

Consensus Revisited. Let us reconsider consensus when $q = 1/2$. The expected payment (obtained according to Theorem 1) of the intuitive simultaneous mechanism is $cn2^n$ since each agent's influence is 2^{1-n} which is the probability that all the other agents declared 1. Consider the equal opportunity mechanism for consensus. The probability that an agent is approached and pivotal equals: Let k be an agent.

$$\sum_i (\mathbf{Pr}\left[\text{k's place is } i\right] \cdot \mathbf{Pr}\left[\text{k is approached in the i-th place}\right] \cdot \mathbf{Pr}\left[\text{k is pivotal}\right].$$

This equals $\frac{1}{n}\sum_i 1/2^{i-1} \cdot 1/2^{n-i} = 1/2^{n-1}$. The probability that an agent is approached is $\frac{1}{n}\sum_i 1/2^{i-1} < \frac{2}{n}$. Therefore, the agent's conditional influence is at least $n/2^n$. According to the payment characterization lemma, each agent approached receives a payment of $v_i < c2^{n+1}/n$. Thus, the expected payment of each agent is bounded by $c2^{n+2}/n^2$ and the overall expected payment is bounded by $c2^{n+2}/n$. This is an improvement by a factor of $\Theta(n^2)$ over the intuitive simultaneous mechanism. Reverting to example of section 1.1 the expected payment decreases from $\$160,000$ to less than $\$26,000$!

5 Approximate Predictions

We saw that exact predictions of functions like consensus or majority may require very large payments. An approximation mechanism for $f(.)$ is a mechanism which gets an accuracy parameter ε and in equilibrium predicts $f(.)$ with accuracy of at least $1-\varepsilon$. This section shows that **every** function can be approximated using a relatively low expected payment. The main problem that needs to be overcome is that when $f(.)$ is almost determined, agents will have very small influence. Avoiding this requires a little care. Due to space constraints we leave only the definition and the main theorem.

Definition 9. (ε-Implementation) *Let $\varepsilon \geq 0$. A (truthful) implementation is called an* ε-implementation *of $f(.)$, if $\mathbf{Pr}\big[f(x) \neq g(x)\big] < \varepsilon$ where $g(x)$ is the actual function that the mechanism implements and the probability is taken over the input distribution ϕ_q.*

Theorem 5. (Approximate predictions of general functions) *Let $\varepsilon < 1/2$. Every prediction function $f(.)$ can be ε-approximated with the following expected payment:*

$$B(f,\varepsilon) = O\left(\frac{cn^2}{\varepsilon(1-2\varepsilon)(1-q_M)\log n}\right).$$

6 Weighted Threshold Functions

The design of low payment mechanisms for non-anonymous functions is an intriguing challenge. A particularly interesting class of functions is the class of weighted threshold functions. These are functions of the form:

$$f_{w,\theta}(x) = \begin{cases} 1 \text{ if } \sum_{i=1}^{n} w_i x_i \geq \theta \\ 0 \text{ otherwise} \end{cases}$$

where w is a vector of non negative numbers and θ a positive threshold. These functions are natural to use in many prediction scenarios. For example when forming an expert committee one could set the experts' weights according to their reputation. We will show that for every two agents, and for every input vector of the other agents, the agent with the higher weight *always* has higher influence. This property indicates that characterize the optimal mechanism for weighted threshold functions may be easier than the general case.

We show that among all the mechanisms which approach the agents in a **fixed** order, the mechanism which approaches the agents by the order of their weights is the best.

Surprisingly, we demonstrate that this is not always the best deterministic mechanism. We do so by introducing a generic de-randomization technique.

Due to space constraints we leave only the definition and the main theorem.

Definition 10. (DWO mechanisms) *A* descending weight order (DWO) mechanism *is a canonical mechanism which approaches the agents serially by a fixed order determined by a descending order of their weights. (The mechanism chooses arbitrarily between agents with the same weight.)*

Theorem 6. *The DWO mechanism dominates every other fixed order mechanism.*

7 Future Research

In this paper we studied a basic class of prediction problems. Real life prediction scenarios are likely to be more complicated. Nevertheless, we believe that the insights we gained here (e.g. the usefulness of a serial approach, randomization, and zero information) are very useful for making such predictions. It would be interesting to extend our setup to several directions: in particular consideration of probabilistic prediction functions, general independent distributions, and general decision making in verifiable situations.

In this paper we assumed that the agents' inputs are independent. An important case is conditionally independent inputs. In this case, the mechanism may be able to use the fact that the agents' inputs are correlated in order to reduce its payment. Another important case is when inputs can be verified at least partially.

When $f(.)$ is polynomially computable, the greedy mechanism can be approximated in polynomial time. Devising polynomial time mechanisms with smaller payment for non anonymous functions is an intriguing challenge.

Finally, it will be interesting to study repeated prediction games where the function $f(.)$ is unknown to the mechanism but can be learnt over time.

Acknowledgment. We thank Irit Dinur, Nati Linial, and Eddy Mayer Wolf for useful discussions. We thank Elad Dukov, Eli Luboshitz, and Inbal Ronen for their comments on earlier drafts of this paper.

References

[1] J. Feigenbaum, L. Fortnow, D. M. Pennock, and R. Sami. Computation in a distributed information market. Proceedings of the 4th ACM Conference on Electronic Commerce (EC), 2003.

[2] D. M. Pennock, S. Lawrence, C. L. Giles, and F. A. Nielsen. The power of play: Efficiency and forecast accuracy in web market games. NEC Research Institute Technical Report num. 168, 2000.

[3] D. M. Pennock, S. Lawrence, F. A. Nielsen, and C. L. Giles. Extracting collective probabilistic forecasts from web games. Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pages 174–183, 2001.

[4] Y. Shoham and M. Tennenholtz. Non-Cooperative Computing: Boolean Functions with Correctness and Exclusivity. *Journal of Theoretical Computer Science (JTCS)*, special issue "Game Theory Meets Theoretical Computer Science".

[5] R. McGrew, R. Porter and Y. Shoham. Towards a General Theory of Non-Cooperative Computing. *Proceedings of the 9th Conference on Theoretical Aspects of Rationality and Knowledge (TARK)*, 2003.

[6] R. Smorodinsky and M. Tennenholtz. Overcoming Free Riding in Multi-Party Computations - The Anonymous Case. Forthcoming, Games and Economic Behavior, 2005

[7] R. Smorodinsky and M. Tennenholtz. Sequential Information Elicitation in Multi-Agent Systems. Proc. of the Twentieth Conference on Uncertainty in Artificial Intelligence (UAI-2004)

[8] J. Kahn, G. Kalai and N. Linial. The Influence of Variables on Boolean Functions. Proceedings of the 29th Annual Symposium on Foundations of Computer Science (ASFSC), Computer Society Press, pages 68-80, 1988.

[9] M. Talagrand. On Russo's approximate 0-1 law. Annals of Probability Vol 22 NO 3, pages 1576-1587, 1994.

[10] G. Kalai and S. Safra. Threshold Phenomena and Influence with some perspectives from Mathematics, Computer Science and Economics. 2004.

[11] A. Gershkov and B. Szentes. Optimal Voting Scheme with Costly Information. Manuscript.

[12] E. Winter. Incentives and Discrimination. *American Economic Review (AER)*, 2004.

[13] A. Mas-Colell, M. Whinston, and J. Green Microeconomic Theory Oxford University Press, 1995.

[14] K.C. Green Forecasting decisions in conflict situations: a comparison of game theory, role-playing, and unaided judgment. *International Journal of Forecasting (IJF)*, Volume 18, Issue 3, Pages 321–344, 2002.

[15] J.S. Armstrong Assessing game theory, role playing, and unaided judgment. *International Journal of Forecasting (IJF)*, Volume 18, Issue 3, Pages 345–352, 2002.

[16] K.Y. Chen, L.R. Fine and B.A. Huberman Forecasting Uncertain Events with Small Groups. Proceedings of the Third ACM Conference on Electronic Commerce (EC), 2001.