

1

Computational Aspects of Prediction Markets

David M. Pennock and Rahul Sami

Abstract

Prediction markets (also known as information markets) are markets established to aggregate knowledge and opinions about the likelihood of future events. This chapter is intended to give an overview of the current research on computational aspects of these markets. We begin with a brief survey of prediction market research, and then give a more detailed description of models and results in three areas: the computational complexity of operating markets for combinatorial events; the design of automated market makers; and the analysis of the computational power and speed of a market as an aggregation tool. We conclude with a discussion of open problems and directions for future research.

1.1 Introduction: What is a prediction market?

Consider the following mechanism design problem called the *information aggregation problem*. Suppose an individual (“the aggregator”) would like to obtain a prediction about an uncertain variable, say the global average temperature in 2020. A number of individuals (“the informants”) each hold different and non-independent sets of information bearing on the outcome of the variable. The goal is to design a mechanism that extracts the relevant information from the informants, aggregates the information appropriately, and provides a collective prediction or forecast. The forecast should ideally be equivalent to the omniscient forecast that has direct access to all the information available to all informants.

A *prediction market*[†] is one mechanism designed to solve the information

[†] Prediction markets are also often referred to as information markets, (Arrow-Debreu) securities markets, contingent claims, contingent contracts, event markets, event futures, event derivatives, and idea futures.

aggregation problem. The aggregator creates a financial *security* whose payoff is tied to the outcome of the variable. For example, he creates a security that pays $\$x$ dollars if the actual global average temperature in 2020 equals x . The aggregator invites the informants to trade the security however they please. For example, global warming proponents should be willing to buy the security at or above prices equal to today’s global average temperature, and global warming skeptics should be willing to sell at those prices.‡ The aggregator can view the trading price of the security as a collective forecast for the expected value of the uncertain variable. In fact, as we shall see in Section 1.2.2.3, in some simplified theoretical settings one can prove that the trading price converges to a *rational expectations equilibrium* that mimics the omniscient forecast.

More importantly, in a broad and diverse number of real-world settings in the lab, in the field, and in practice, prediction markets seem to yield equal or better forecasts than other methods of information aggregation. Researchers have proposed using prediction markets to help scientists, policymakers, decision makers, the government, and the military. Several companies—from established brands like Google, Microsoft, and Yahoo! to startups like CrowdIQ, InklingMarkets, and NewsFutures—are experimenting with prediction market services in the private sector. The growth of the field is reflected and fueled by a wave of popular press articles and books on the topic, most prominently Surowiecki’s “The Wisdom of Crowds”.

In this chapter, we focus on algorithmic challenges and constraints associated with implementing a prediction market mechanism. We discuss three areas in which computational constraints are important:

- Effective prediction markets often need to handle combinations of different events or contingent events. However, the number of contingent events grows exponentially in the number of base events. In this situation, the basic functions of listing securities and clearing markets can become computationally intractable. In Section 1.3, we present results on the computational complexity of operating combinatorial markets.
- In order to increase trading volume, a prediction market operator often acts as a *market maker* who is always ready to trade. However, in order to limit the exposure of the market maker, it is essentially that the market maker adjusts its bid and ask prices after every trade. In Section 1.4, we describe two new designs to automate the price updating process in a way that limits exposure while encouraging informed traders to trade.
- When different traders have complementary information about the value

‡ For simplicity, we ignore the time value of money.

of a security, the market itself ideally performs a computational function: The final trading price should reflect an aggregate of all the traders' initial information. In Section 1.5, we present a simple market model and analyze its computational properties. We derive positive and negative results on when the market will converge to the ideal price, as well as bounds on a measure of convergence time.

In Section 1.2, we set up the problem formally and survey the academic literature on prediction markets.

1.2 Background

1.2.1 Setup and Notation

In this section we formally pose the aggregation problem that prediction markets are designed to address. We begin by introducing a fairly standard model of uncertainty and distributed information.

Definition 1.1 Partition model of knowledge: There is a set Ω of possible *states of the world*. At any point of time, the world is in exactly one state $\omega \in \Omega$, but agents do not necessarily know the true state of the world. However, each agent i may have partial information about the true state. Agent i 's *information* is represented by a *partition* π_i of Ω ; that is, π_i is a collection $\{\pi_{i1}, \pi_{i2}, \dots, \pi_{ik}\}$ of subsets of Ω such that the different subsets are disjoint and the union of all subsets is Ω . The semantic interpretation is that i can distinguish two states in different subsets π_{i1}, π_{i2} of her partition π_i , but cannot distinguish between two states in the same subset of the partition. In particular, agent i knows in which subset of her partition the true state of the world lies, but does not know which member of that subset is the true state. Given n agents $1, 2, \dots, n$, their combined information $\hat{\pi}$ is the *coarsest common refinement* of the partitions $\pi_1, \pi_2, \dots, \pi_n$.

The partition model is often augmented with the assumption that there is a common *prior probability distribution* $\mathcal{P} \in \Delta(\Omega)$ which captures the probability that all agents assign to different states before receiving any information. Once agents obtain their partial information, their *posterior* beliefs follow by conditioning on their information—that is, by restricting the prior to the subset of their partition in which the true state lies.

A *forecast* is an estimate of the expected value of some function $f(\omega)$, where f is a commonly known (deterministic or stochastic) function of the state of the world. A special type of function $f : \Omega \rightarrow \{0, 1\}$ called an *event* equals one for a particular subset of Ω and zero everywhere else. A *joint*

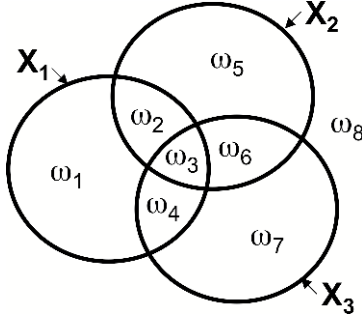


Fig. 1.1. Partition model of knowledge. In this example, the set Ω of states of the world contains eight mutually-exclusive and exhaustive states: $\omega_1, \omega_2, \dots, \omega_8$. Subsets of states like X_1 , X_2 , and X_3 are called *events*. Suppose that agent i can distinguish between states in X_i and states not in X_i , but cannot further distinguish among states. For example, agent 1's partition π_1 is $\{\{\omega_1, \omega_2, \omega_3, \omega_4\}, \{\omega_5, \omega_6, \omega_7, \omega_8\}\}$. In this simple example, the coarsest common refinement of the three agents' partitions is $\hat{\pi} = \Omega$, meaning that the agents' combined information is always sufficient to precisely identify the true state. Often, we may consider the events X_i as the most basic elements of the model, with the ω_i being the implied product space of these base event outcomes. For example, ω_4 in the figure is explicitly indexed as $\omega_{X_1 \bar{X}_2 X_3}$: the future state where X_1 is true, X_2 is false, and X_3 is true.

forecast is a joint probability distribution over the values of a number of functions $f_1(\omega), f_2(\omega), \dots$

On its own, an agent's best forecast uses its posterior distribution over Ω , but ignores information that might be obtained via interaction with other agents. The *omniscient forecast* uses the posterior distribution conditioned on all information available to all agents, or \mathcal{P} restricted to the subset of $\hat{\pi}$ in which the true state lies.

In reality, each agent's information is private knowledge that's not directly accessible to any one entity. Thus information aggregation is a problem of mechanism design (see Chapter 9). The goal is to produce a mechanism that incentivizes the agents to reveal their information such that, in equilibrium, the mechanism produces a forecast as close as possible to the omniscient forecast.

A prediction market is one type of information aggregation mechanism. The market contains financial securities whose payoffs are functions of the state of the world. In the simplest case, the market contains a security paying off $f(\omega)$ dollars in state ω . Thus agents are incited through the prospect of financial gain to reveal information bearing on the expected value of $f(\omega)$, and the equilibrium price reached by a number of interacting agents can be

viewed as a collective forecast. As we shall see in Section 1.5, even when a single forecast is sought, multiple securities might be required to ensure convergence to equilibrium. In Section 1.3 we explore the computationally challenging case of setting up a market to yield a joint forecast.

1.2.2 Survey of the Field

The field of prediction markets is largely an empirical science, and much of the academic literature focuses on laboratory and field experiments testing the accuracy of predictions in a variety of settings. However, a prediction market is operationally no different than a standard financial market, so a large amount of economic and financial theory applies.

1.2.2.1 What and how: Instruments and mechanisms

A prediction market can be designed to elicit a forecast for any type of random variable or set of variables. For example, the variable can be binary (“will a Republican win the next US Presidential election?”), discrete (“who will win the next US Presidential election? A Democrat, a Republican, or someone else?”), continuous (“what will the global average temperature be in 2020?”), or a joint space of any combination of the above.

Beyond “what” is being traded, there are a variety of different mechanisms specifying “how” the securities are traded, including a call market auction, continuous double auction, continuous double auction with market maker, bookmaker, parimutuel market, and combinatorial versions of the above, all of which have some empirical record of success.

In a call market auction, all bids are collected over time, then processed together in large batches. The clearing price can be the m th lowest price, the $m + 1$ st lowest price, or somewhere in between, where m is the number of sellers. A continuous double auction is a continuous version of a call market, where as soon as any trade is acceptable to any two bidders, the trade is immediately executed, usually at the bid price of the least recent bidder. A market maker or bookmaker is a price maker who is nearly always willing to accept both buy and sell orders at some stated (but changing) prices. In a parimutuel market, players compete in a wagering game to earn as large a portion as possible of the total amount of money wagered by all players.

1.2.2.2 Examples and evaluations

A prediction market cannot surface information that does not exist or is unknown, so the accuracy of a prediction market can only be evaluated in comparison to other information aggregation or forecasting methods. The

central empirical question is whether a prediction market aggregates or summarizes information more accurately than other methods.

One of the most cited and most successful prediction markets is the Iowa Electronic Market (IEM). Since 1988, IEM has been operating real-money prediction markets, mostly on the outcomes of political elections. Empirically, on average the market's predictions are more accurate and less volatile than political opinion polls, especially in large US elections. The markets react to new information quickly, sometimes within minutes, and often before the new information becomes widespread. The markets are accurate despite documented evidence that individual traders are often biased, irrational, and make mistakes. Several IEM publications support a theory that accuracy derives not from average traders, but from *marginal traders*. Marginal traders are more active, less biased, more successful, and are price makers rather than price takers. As long as a few good marginal traders exist, the the market as a whole remains accurate despite the poor traders.

Options, futures, and other financial derivatives are contracts whose payoff is a function of some underlying uncertain variable. For example, the payoff of a stock option with strike price k is $\max[0, s - k]$, where s is the price of the corresponding stock at some future date. Sports betting markets can also be viewed and analyzed as prediction markets. Several empirical studies verify that derivative prices and sports betting odds constitute accurate forecasts for their underlying variables.

Even play-money markets show a surprising ability to aggregate information. Studies of market games like the Hollywood Stock Exchange, News-Futures, and the Foresight Exchange report accuracies equal to or better than expert opinions and, remarkably, sometimes on par with equivalent real-money prediction markets.

Experimental economists have tested the aggregation properties of prediction markets in laboratory settings. The experimenter sets up the forecasting problem and carefully controls the information each participant receives. A number of experimental designs reveal when market aggregation seems to work and when it doesn't. Generally, given enough securities and enough practice, traders in the lab often converge to prices close to the omniscient forecasts. Researchers have devised and tested methods for achieving accurate results across as many forecast variables as possible with as few participants as possible.

Economists have also run field tests of markets used to forecast quantities of interest to an organization. For example, a market was tested at Hewlett Packard to project the company's sales volume for particular products. Generally, the market predictions were superior to the official HP

forecasts. Other companies, including Microsoft and Google, are now running similar internal prediction markets.

1.2.2.3 Theoretical underpinnings

There is a fundamental difference between a market for a financial security and a market for a consumer product: the security has no direct consumption value to potential buyers. Buyers only want to buy the security because they believe they can later resell it or cash it out for a higher price. This simple observation invalidates the classical model of demand, in which each trader has a fixed demand curve that describes the quantity demanded at each price. The market provides information about other traders' knowledge and beliefs, which may lead a trader to change her beliefs about the future value of the security. In this manner, *the market prices can lead to changes in the traders' demand curves!* This led to the development of a new theory, the theory of *rational expectations*, that seeks to understand this latter kind of market. The cornerstone of this theory is a new equilibrium concept, the *rational expectations equilibrium*. Intuitively, a rational expectation equilibrium price is a market-clearing price such that traders will not want to change their trades *even after observing the price itself*.

Rational expectations Consider the model of Section 1.2.1: an uncertain world with possible states Ω , and n traders trading in a market for some good. Let $v_i(q_i, \omega)$ denote the ultimate value of q_i units of the good to trader i in state ω . The traders are partially informed: let π_i denote trader i 's private information, and assume there is a common prior distribution \mathcal{P} . Further, we assume that all traders are risk-neutral Bayesians. In order to simplify the exposition, we consider the special case in which the $\hat{\pi} = \Omega$, so the combined information of all agents is sufficient to pinpoint the true state. The equilibrium price is not a simple number as in the case of the competitive equilibrium; instead, it is a mapping $P^* : \Omega \rightarrow \mathfrak{R}$ that maps a state of the world to a price.

Definition 1.2 A *rational expectations equilibrium* is a mapping $P^* : \Omega \rightarrow \mathfrak{R}$ such that in every state ω , if every trader conditions her demand (or supply) on her private information π_i as well as the price $P^*(\omega)$, the market will clear at a price of exactly $P^*(\omega)$. In other words, it is a self-fulfilling correspondence from states to prices.

This definition is subtle, and needs to be reasoned through carefully. Consider an arbitrary non-constant mapping P from states to prices. Then, by

observing the price $P(\omega)$, an agent who knew the mapping could immediately rule out some states of the world: those that would have resulted in a different price. Thus, any mapping P induces a partition π_P such that anyone observing $P(\omega)$ knows π_P in addition to her initial information. Now, trader i 's effective demand curve in state ω will be given by her expected value for the item conditioned on both the price and her private information: $\tilde{v}_i(q_i, \omega) = E[v(q_i, \omega) | \pi_i(\omega), P(\omega)]$. Given the demand and supply curves for the n agents, it is possible to calculate a clearing price $\tilde{v}(\Omega)$. The price mapping P would be a rational expectations equilibrium iff $\tilde{v}(\omega) = P(\omega)$ for all ω . In other words, it is rational for the agents to believe in a price mapping P only if all agents believing in that mapping and acting accordingly would lead to the prices predicted by P .

Researchers have shown the existence of rational expectations equilibria in economies with asymmetric information under fairly general conditions on the value functions $v_i(\cdot, \cdot)$. Further, it has been shown that under generic conditions, these economies admit “fully-revealing” rational expectations equilibria: price correspondences $P^*(\cdot)$ such that $P^*(\omega_1) \neq P^*(\omega_2)$ whenever $\omega_1 \neq \omega_2$. In this case, it follows that the price reveals the combined information of all traders, *i.e.*, $\pi_{P^*} = \hat{\pi}$, the full-information partition. This leads to startling, and sometimes counter-intuitive, consequences; we discuss some of these in subsequent sections. We note, however, that the rational expectations literature has been criticized because the definition of a rational expectations equilibrium says nothing about how traders might learn and agree on the equilibrium price mapping P^* . In applying this concept, it is important to keep this in mind, and take the price formation process into account when possible.

Efficient market hypothesis and no-trade theorems The existence of fully revealing equilibria has led researchers to propose the “efficient market hypothesis” . The strong form of this hypothesis states that a security’s market price fully reflects all the information relevant to its value. The efficient market hypothesis, with its roots in rational expectations theory, provides a theoretical foundation for why prediction markets are likely to be effective: In a situation in which many traders have a small amount of private information about an event, it states that the prediction market price will reflect the combined information of all traders.

One of the most counter-intuitive results of rational expectations theory is the existence of *no-trade theorems*. The key observation is that, in a fully revealing rational expectations equilibrium, the price information captures every agent’s private information. Thus, in a fully revealing equilibrium,

all agents are conditioning their beliefs on identical information, and hence have identical posterior beliefs. It follows that all agents assign the same expected value to the security, and hence, there will not be any trade in equilibrium. This reasoning can be extended to show that no two rational agents will want to trade with each other even if they are not initially in equilibrium, because the mere willingness of the other party to trade at a given price reveals information that leads to an equilibrium. Several variants of this result, under different conditions, have been shown.

Thus, we seem to have a paradoxical situation in which the final price reflects all the traders' information, but the traders would never want to trade so there is no way for their information to get into the prices! However, the no-trade results are very sensitive to the precise conditions specified – risk-neutrality and common knowledge that all traders are completely rational Bayesians – and even tiny perturbations of these conditions invalidate them. In practice, there are several reasons that can lead an informed trader to expect a profit from trade, such as the existence of irrational traders, traders who are trading to hedge risks, traders who trade for liquidity reasons, or a market-maker who is subsidizing the market.

1.3 Combinatorial Prediction Markets

Up to this point, we have concentrated on the economic, strategic, and statistical properties of prediction markets. We now turn our attention to the computational problems that arise in the study of prediction markets. In this section, we consider *combinatorial* markets. These are markets in which the state space is the product space of a number of *base events*. Here, we consider state spaces generated by *Boolean events*: propositions such as, “the price of gasoline is greater than \$3” that may be either true or false in the future world. Suppose there is some finite set \mathcal{E} of base events, and further, suppose that these events are linearly independent in the sense that the value (**true** or **false**) of any event cannot be determined with certainty even if the value of all other events is known. Then, the state space Ω is of size $2^{|\mathcal{E}|}$, with each state corresponding to a particular assignment of values to the individual events. We use the symbols X_1, X_2, X_3, \dots to denote the individual Boolean events in \mathcal{E} .

Let S_ω be a security that pays \$1 if the eventual state is ω , and pays \$0 otherwise. Classic results on market equilibrium show that a market can be guaranteed to be efficient if it is possible for a trader to express her desire for any such S_ω . This does not necessarily mean that the securities S_ω have to be directly traded in the market, as long as the market has a set of securities

such that a trader could construct a portfolio with payoff similar to any S_ω she desires. Such a market is called a *complete* market. Unfortunately, any complete market must have at least $2^{|\mathcal{E}|}$ securities; if the number of base events is large, even listing all the securities may be impossible!

However, this does not mean that it is impossible to achieve efficient hedging or information aggregation in practice. There may be many fewer than $2^{|\mathcal{E}|}$ combinations of events that traders actually care about, or have specific information about. This raises the following questions: (1) Is there a “natural” representation such that realistic events, securities, and buy/sell orders can be represented succinctly? (2) Given orders in this representation, is it possible to identify and execute possible trades?

The underlying structure of the state space can be exploited through the use of prediction markets with expressive bidding languages. We distinguish between two forms of expressivity: *combined orders* and *compound orders*.

A combined order allows the trader to specify a collection of securities he or she would like to trade together as a bundle, with limit prices specified for each component security. If the trader cannot obtain all of the securities at prices equal to or better than the specified limits, then the trader prefers not receive any of the securities. This form of expressivity reduces so-called *execution risk*, where during the course of carrying out a planned series of transactions, the prices of some securities change, thereby reducing or reversing the utility of the earlier trades. If there are $|\mathcal{E}|$ Boolean event securities, then traders can place a combined order for any of the $2^{|\mathcal{E}|}$ possible bundles (subsets) of the securities. When combined orders are allowed, the auctioneer problem is essentially the same as in the combinatorial auction scenario (see Chapter 11). One distinction is that, while bids in combinatorial auctions are generally considered indivisible, bids in a securities market often can be considered divisible, thus simplifying the matching problem. The auctioneer problem of matching combined orders in a securities market is also called *combined value trading*.

A compound order allows the trader to speculate on any compound Boolean expression involving a set \mathcal{E} of base events. If there are $|\mathcal{E}|$ base events, then there are $2^{|\mathcal{E}|}$ possible combinations of outcomes of those events, and there are $2^{2^{|\mathcal{E}|}}$ distinct subsets of those combinations expressible using boolean formulas. For the remainder of this section, we will focus on compound orders, a strict superset of combined orders.

1.3.1 Compound Prediction Markets

We now describe a concrete representation for compound order securities. The securities are based on Boolean formulas over the set of propositions \mathcal{E} ; given a formula ϕ , we have a security that pays \$1 *iff* ϕ is true in the eventual state. More generally, we allow *conditional* securities $S_{\phi|\psi}$ based on two formulas ϕ, ψ ; this is interpreted as, “Make a payoff according to ϕ , conditional on ψ being true”. In other words, the owner of security $S_{\phi|\psi}$ is paid \$1 if both ϕ and ψ are true, paid \$0 if ψ is true but ϕ is false, and the security is cancelled (and any money the owner paid for it is refunded) *iff* ψ is false.

1.3.1.1 Orders

Agents place orders, denoted o , of the form “ q units of $S_{\phi|\psi}$ at price p per unit”, where $q > 0$ implies a buy order and $q < 0$ implies a sell order. We assume agents submitting buy (sell) orders will accept any price $p^* \leq p$ ($p^* \geq p$). We distinguish between *divisible* and *indivisible* orders. Agents submitting divisible orders will accept quantity αq for any $0 < \alpha \leq 1$. Agents submitting indivisible orders will accept only exactly q units, or none.

Every order o can be translated into a payoff vector Υ across all states $\omega \in \Omega$. The payoff $\Upsilon^{(\omega)}$ in state ω is $q \cdot 1_{\omega \in \psi} (1_{\omega \in \phi} - p)$, where $1_{\omega \in E}$ is the indicator function equaling 1 *iff* $\omega \in E$ and zero otherwise. Let the set of all orders be $\mathcal{O} = \{o_i\}$ and the set of corresponding payoff vectors be $\mathcal{P} = \{\Upsilon_i\}$.

1.3.1.2 The matching problem

The auctioneer’s task, called the *matching problem*, is to determine which orders to accept among all orders $o \in \mathcal{O}$. Let α_i be the fraction of order o_i accepted by the auctioneer (in the indivisible case, α_i must be either 0 or 1; in the divisible case, α_i can range from 0 to 1). If $\alpha_i = 0$, then order o_i is considered rejected and no transactions take place concerning this order. For accepted orders ($\alpha_i > 0$), the auctioneer receives the money lost by bidders and pays out the money won by bidders, so the *auctioneer’s payoff vector* (or *surplus vector*) is:

$$\Upsilon_{\text{auc}} = \sum_{\Upsilon_i \in \mathcal{P}} -\alpha_i \Upsilon_i.$$

Assume that the auctioneer wants to choose a set of orders so that it is guaranteed not to lose any money in any future state, but that the auctioneer does not necessarily insist on obtaining a positive benefit from the transaction (i.e., the auctioneer is content to break even).

Definition 1.3 (Indivisible matching problem) Given a set of orders \mathcal{O} , does there exist $\alpha_i \in \{0, 1\}$ with at least one $\alpha_i = 1$ such that $\forall \omega, \Upsilon_{\text{auc}}^{(\omega)} \geq 0$? In other words, does there exist a nonempty subset of orders that the auctioneer can accept without risk? \square

Example 1.4 (Indivisible order matching) Suppose $|\mathcal{E}| = 2$. Consider an order to buy one unit of $S_{X_1 X_2}$ at price 0.4 and an order to sell one unit of S_{X_1} at price 0.3. The corresponding payoff vectors are:

$$\begin{aligned} \Upsilon_1 &= \langle \Upsilon_1^{\langle X_1 X_2 \rangle}, \Upsilon_1^{\langle X_1 \bar{X}_2 \rangle}, \Upsilon_1^{\langle \bar{X}_1 X_2 \rangle}, \Upsilon_1^{\langle \bar{X}_1 \bar{X}_2 \rangle} \rangle \\ &= \langle 0.6, -0.4, -0.4, -0.4 \rangle \\ \Upsilon_2 &= \langle -0.7, -0.7, 0.3, 0.3 \rangle \end{aligned}$$

The auctioneer's payoff vector (the negative of the component-wise sum of the above two vectors) is:

$$\Upsilon_{\text{auc}} = -\Upsilon_1 - \Upsilon_2 = \langle 0.1, 1.1, 0.1, 0.1 \rangle.$$

Since all components are nonnegative, the two orders match. The auctioneer can process both orders, leaving a surplus of \$0.1 in cash and one unit of $S_{X_1 \bar{X}_2}$ in securities. \square

Now consider the divisible case, where order can be partially filled.

Definition 1.5 (Divisible matching problem) Given a set of orders \mathcal{O} , does there exist $\alpha_i \in [0, 1]$ with at least one $\alpha_i > 0$ such that $\forall \omega, \Upsilon_{\text{auc}}^{(\omega)} \geq 0$? \square

The matching problems defined above are decision problems: the task is only to show the existence or nonexistence of a match. We could additionally seek to maximize some objective function—like trading volume or auctioneer expected profit—to choose the best among all possible matches. Here, we restrict our attention to the decision problem formulations.

1.3.1.3 The computational complexity of matching

In this section we examine the computational complexity of the auctioneer's matching problem. Here n is the size of the problem's input, including descriptions of all the buy and sell orders. We also assume that n bounds the number of base securities. We consider four cases based on two parameters:

- (i) Whether to allow divisible or indivisible orders.
- (ii) The number of securities. We consider two possibilities: (a) $O(\log n)$ base securities yielding a polynomial number of states, or (b) $\Theta(n)$ base securities yielding an exponential number of states.

Theorem 1.6 *The matching problem for divisible orders is*

- (i) *computable in polynomial-time for $O(\log n)$ base securities.*
- (ii) *co-NP-complete for unlimited securities.*

Proof

Small number of securities with divisible orders We can build a linear program based on Definition 1.5. We have variables α_i . For each i , we have $0 \leq \alpha_i \leq 1$. and for each state ω in Ω we have the constraint

$$\text{Payment}(\omega) = \sum_i -\alpha_i \Upsilon_i^{(\omega)} \geq 0.$$

Given these constraints we maximize $\sum_i \alpha_i$. A set of orders has a matching exactly when $\max \sum_i \alpha_i > 0$. With $O(\log n)$ base securities, we can solve this linear program in polynomial time. Note however that this approach may not find matchings that have precisely zero surplus.

Large number of securities with divisible orders With unlimited base securities, the linear program given in Section 1.3.1.3 has an exponential number of constraint equations. Each constraint is short to describe and easily computable given ω . Let $m \leq n$ be the total number of buy and sell orders. By the theory of linear programming, an upper bound on the objective function can be forced by a collection of $m + 1$ constraints. So if no matching exists there must exist $m + 1$ constraints that force all the α_i to zero. In nondeterministic polynomial-time we can guess these constraints and solve the reduced linear program. This shows that matching is in co-NP.

To show co-NP-completeness we reduce the NP-complete problem of Boolean formula satisfiability to the nonexistence of a matching. Fix a formula ϕ . Let the base securities be the variables of ϕ and consider the single security S_ϕ with a buy order of 0.5. If the formula ϕ is satisfiable then there is some state with payoff 0.5 (auctioneer payoff -0.5) and no fractional unit of security S_ϕ is a matching. If the formula ϕ is not satisfiable then every state has an auctioneer's payoff of 0.5 and a single unit of S_ϕ is a matching. \square

For *indivisible* orders, the matching problem turns out to be even harder to solve. We state the following result; due to space restrictions, we do not reproduce the proof here.

Theorem 1.7 *The matching problem for indivisible orders is*

- (i) *NP-complete for $O(\log n)$ base securities.*
- (ii) *Σ_2^P -complete for unlimited securities.*

1.3.2 Compact Prediction Markets

Compound orders are very general: traders can submit orders for any boolean expression of base events. Computational limits aside, a market system supporting compound orders effectively implements a complete securities market, as defined above, meaning that all possible mutually agreeable transactions can proceed, supporting a Pareto optimal and economically efficient allocation of securities.

Of course, computational limits are a real practical barrier; matching compound orders can easily become intractable. By limiting the full expressivity of compound orders, computational complexity can be reduced.

One natural restriction takes advantage of any (conditional) independence relationships among base events. Suppose that the statistical dependency structure of the base events is encoded as a Bayesian network. That is, the joint probability distribution over the base events can be factored as follows:

$$\Pr(X_1 X_2 \cdots X_{|\mathcal{E}|}) = \prod_{k=1}^{|\mathcal{E}|} \Pr(X_k | \mathbf{pa}(X_k)),$$

where $\mathbf{pa}(X_k)$ is a set of base events with index less than k called X_k 's *parents*. The factorization can be depicted as a directed acyclic graph with nodes representing base events and edges from each event in $\mathbf{pa}(X_k)$ to X_k representing direct conditional dependencies.

Now restrict trading to conditional securities of the form $S_{X_j | \mathbf{pa}(X_j)}$, one for each conditional probability $\Pr(X_j | \mathbf{pa}(X_j))$ in the Bayesian network. Each event X_j with $|\mathbf{pa}(X_j)|$ parents corresponds to $2^{|\mathbf{pa}(X_j)|}$ securities, one for each possible combination of outcomes of events in $\mathbf{pa}(X_j)$. A securities market structured in this way contains $O(|\mathcal{E}| \cdot 2^{\max |\mathbf{pa}(X_j)|})$ securities, which can be considerably fewer than the $2^{|\mathcal{E}|}$ securities required for a complete market, if $\max |\mathbf{pa}(X_j)| \ll |\mathcal{E}|$. Call such a market a *BN-structured market*.

Although the need for $2^{|\mathcal{E}|}$ securities cannot be relaxed if one wants to guarantee completeness in all circumstances, there are some restrictive conditions under which smaller BN-structured securities market may be *operationally complete*, meaning that its equilibrium is Pareto optimal *with respect to the traders involved*. In particular, if all traders' risk-neutral independencies agree with the independencies encoded in the market structure, then the market is operationally complete. For collections of agents all with constant absolute risk aversion (negative exponential utility for money), agreement on *Markov independencies* is sufficient for operational completeness.

1.4 Automated Market Makers

The standard way to organize a market is as a continuous double auction, in which traders arrive asynchronously and place their orders, and a trade takes place if a buyer quotes a higher price than a seller who is present at the same time. In a prediction market organized in this way, a speculator with private information about the security would have to submit her order and wait for another trader to place a matching order.

There are two problems with this scenario. First, the informed trader may not be willing to wait indefinitely for a partner to trade with. If there are few potential traders, they may never even enter the market because they do not expect to find a trading partner. This is the *thin market* problem: a “chicken and egg” scenario where few traders care to participate because other traders are scarce, leading to a potential breakdown of the market. The thin market problem can be especially severe in a combinatorial market because each trader’s attention is divided among an exponential number of choices, making the likelihood of a match between traders seem very remote. Second, an informed trader may not want to reveal her willingness to trade (at a given price), because this may tip off other traders, and may prevent her from making a profit. This effect is related to the no-trade theorems discussed in 1.2.2.3, and arises because traders are essentially playing a zero-sum game with each other. Both problems can reduce the incentives for traders to participate, thus reducing the informativeness of prices.

An alternative to using a double auction mechanism is for the market to include a *market maker*. A market maker is an agent who is always ready to trade. Typically, a market maker posts bid and ask prices (which may be identical); then, a seller who is willing to sell at the bid price (or a buyer who is willing to pay the ask price) can trade with the market maker. The market maker may later resell the securities it bought to a buyer. In this way, the market maker can effectively engineer a trade between a buyer and a seller who arrive at different times and do not wait.

Of course, one side effect of having a market maker is that the market operator could potentially make a loss. This is not necessarily a negative property; in essence, it is a way of injecting subsidies into the market. The no-trade theorems no longer apply to a market with subsidies, so informed speculators can rationally expect to profit from their trade. However, it is important that the loss be predictable or bounded. In order to achieve this, the bid and ask prices must be adjusted in a systematic way after every trade; the new prices are computed by an *automated market maker*.

An ideal automated market maker should satisfy three properties: (1) it

should run a predictable or bounded loss; (2) informed traders should have an incentive to trade whenever their information would change the price; and (3) after any trade, computing the new prices should be a tractable problem. In this section, we describe two new microstructures for prediction markets that effectively function as automated market makers, and appear to have all these properties.

1.4.1 Market Scoring Rules

Hanson shows how any proper scoring rule, or payment scheme designed to elicit truthful reporting of probabilities, can be converted into an automated market maker. The market maker can be thought of as a sequential shared version of the scoring rule, as we describe later. First, we describe the market maker algorithm in a more conventional light.

Suppose the market contains $|\Omega|$ mutually exclusive and exhaustive securities. Let q_j be the total quantity of security j held by all traders combined, and let \vec{q} be the vector of all quantities held. The market maker utilizes a *cost function* $C(\vec{q})$ that records the total amount of money traders have spent as a function of the total number of shares held of each security. A trader who wants to purchase δ shares of security j must pay $C(q_1, \dots, q_j + \delta, \dots, q_{|\Omega|}) - C(\vec{q})$ dollars. More generally, a trader who wants to buy or sell any bundle of securities (*i.e.*, any *combined order* or *compound order*, as defined in Section 1.3) such that the total number of outstanding shares changes from \vec{q}_{old} to \vec{q}_{new} must pay $C(\vec{q}_{new}) - C(\vec{q}_{old})$ dollars. Negative quantities encode sell orders and negative “payments” encode sale proceeds earned by the trader. At any time, the going price of security j is $\partial C / \partial q_j$, the cost per share for purchasing an infinitesimal quantity. The full cost for purchasing any finite quantity is the integral of price evaluated from \vec{q}_{old} to \vec{q}_{new} , or $C(\vec{q}_{new}) - C(\vec{q}_{old})$. Once the true outcome becomes known, the market maker pays \$1 per share to traders holding the winning security.

Deriving the cost function associated with a particular scoring rule is straightforward if tedious. The cost function corresponding to the logarithmic scoring rule is:

$$C(\vec{q}) = b \ln \left(\sum_j e^{q_j/b} \right)$$

and the price function is $\partial C / \partial q_j = e^{q_j/b} / \sum_k e^{q_k/b}$. The free parameter b controls both the market maker’s risk of loss and the effective liquidity of the market. One can show that the maximum possible loss incurred by the market maker is $b \ln |\Omega|$. But a larger b also means that more shares can

be purchase at or near the current price without driving up the price too much, a measure of market liquidity and depth. The logarithmic scoring rule market maker has been implemented in several real-world settings with success, including at InkingMarkets, Net Exchange, and Microsoft.

The cost function corresponding to the quadratic scoring rule is:

$$C(\vec{q}) = \frac{\sum_j q_j}{|\Omega|} + \frac{\sum_j q_j^2}{4b} - \frac{(\sum_j q_j)^2}{4b|\Omega|} - \frac{b}{|\Omega|}.$$

The quadratic scoring rule market maker is likely not of much practical interest. The market maker only allows traders to buy a small fixed number of shares of any security. Moreover, as soon as one upper limit is reached on any security, the market maker cannot accept buy orders for other securities. In contrast, the logarithmic scoring rule market maker can accept arbitrarily large quantities of buy or sell orders.

As mentioned, a market scoring rule market maker can be viewed as a sequential shared version of a scoring rule. Conceptually, the market maker begins by setting prices equal to an initial probability estimate. The first trader to arrive agrees to (1) pay the market maker the scoring rule payment associated with the market maker's probability estimate, and (2) receive the scoring rule payment associated with the trader's own probability estimate. Myopically, this modified scoring rule still incents the trader to reveal her true probability estimate. The final trader pays the scoring rule payment owed to the second-to-last trader and receives a scoring rule payment from the market maker. The market maker's loss is bounded by the maximum possible payment to the final trader minus the payment from the first trader. One can show that the more conventional cost function formulation of the market maker is equivalent to the sequential shared scoring rule formulation.

1.4.2 Dynamic Parimutuel Markets

A *parimutuel* game is a wagering game where players compete to earn as large a portion as possible of the total pool of money wagered by all players. Again consider a set Ω of mutually exclusive and exhaustive outcomes. Players wagers money on the outcome(s) of their choice. When the true outcome is revealed, players who wagered on the correct outcome split the total pool of money in proportion to the amount they bet. In a sense, the cost of purchasing an equal share of the winnings associated with any outcome is always a constant, say \$1. A dynamic parimutuel market is a dynamic-cost variant of the parimutuel wagering game. As before, traders compete for a share of the total money wagered, however the cost of a single share varies

dynamically according to a cost function, thus allowing traders to sell their shares prior to the determination of the outcome for profits or losses. From a trader's perspective, the mechanism acts as a market maker.

A particularly natural cost function is the *share-ratio cost function*, which equates the ratio of prices of any two outcomes with the ratio of number of shares outstanding for the two outcomes. The share-ratio cost function is

$$C(\vec{q}) = \kappa \sqrt{\sum_j q_j^2},$$

where κ is a free parameter. The corresponding price function is $p_j = \kappa q_j / \sqrt{\sum_k q_k^2}$. This cost function is the unique dynamic parimutuel cost function satisfying the ratio constraint $p_j/p_k = q_j/q_k$ for all j and k . Setting $\kappa = 1$ yields a natural version where the price of each outcome is always less than one, and the payoff per share of each outcome is always greater than one. The share-ratio cost function is arbitrage-free and ensures that wagers on the correct outcome can never lose money. The market maker initiates the game with an allocation of shares \vec{q} and a corresponding $C(\vec{q})$ dollars, reflecting the market maker's maximum risk of loss.

Besides the different form of the cost function, the main difference between a market scoring rule market maker and a dynamic pari-mutuel market maker is that the former pays a fixed \$1 per share to winning shareholders while the latter pays an equal portion of the total amount wagered to winning shareholders. Because of the added uncertainty surrounding the payoff per share, trading strategies in a dynamic pari-mutuel market are more complicated, and the interpretation of the price as a forecast is less direct. On the other hand, as a gambling game, the added uncertainty may appeal to risk seeking traders.

1.5 Distributed Computation through Markets

Sections 1.3 and 1.4 concerned algorithmic components of the operation of a prediction market. In this section, we turn that viewpoint inside out, and study the system of market and traders as a computational device (that is perhaps a part of a larger computation)! We construct and analyze a simple model of a prediction market in order to gain insight into two fundamental properties of any computational device: what can it compute? and, how fast does the computation run?

Where is this computation taking place? The traders use their private information to attempt to make profitable trades. Importantly, they observe the market clearing price (or the actual sequence of trades), and *update their*

beliefs about the security value. The computation of the market as a whole occurs through the traders' belief-updating processes; this is where a trader takes a signal (the market price) that reflects some information about other traders, and combines it logically with her own private information.

The process by which the market prices adjust is important for another reason: Recall from section 1.2.2.3 that the rational expectations equilibrium definition does not address the issue of how traders reach the equilibrium price correspondence. We shall see that this can be problematic: With a plausible belief-updating process, the market prices may sometimes get stuck at a noninformative equilibrium, even though a fully-revealing equilibrium exists. Thus, we need a better understanding of the dynamics of the price adjustment process. The following model provides some insight.

1.5.1 Boolean Market Model

We model a very simple class of elementary computation problems – computing a Boolean function – and study what can be computed *with a single security*. Initially, suppose there are n traders, each with a single bit x_i of private information; we use \mathbf{x} to denote the vector (x_1, \dots, x_n) . This model can be translated to a partition model as described in section 1.2.1: The state space is $\Omega = \{0, 1\}^n$, and each agent i initially has a partition $\pi_i = \{\{\mathbf{x} \in \Omega | x_i = 0\}, \{\mathbf{x} \in \Omega | x_i = 1\}\}$ with two components. We are interested in learning the value of a Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ of the combined information \mathbf{x} . To do this, we set up a market in a security F that will pay \$1 if $f(\mathbf{x})$ is ultimately revealed to be 1, and \$0 otherwise. The form of f (the description of the security) is common knowledge among agents. We sometimes refer to the x_i as the *input bits*. At some time in the future after trading is completed, the true value of $f(\mathbf{x})$ is revealed. Note that the traders' combined information is enough to determine the exact value of $f(\mathbf{x})$; thus, if the market is truly efficient, we expect its equilibrium trading price to be equal to $f(\mathbf{x})$.

To have a model that permits analysis, we next need to specify how the market prices are formed, and how the agents bid in the market and react to market information.

1.5.2 Bid Format and Price Formation

Continuous double auctions are complex systems, and there is no standard way to analytically model the price formation process; we use the following linear model which loosely captures the nature of the market, and permits

analysis. The market proceeds in synchronous rounds. In each round, each agent i submits a *bid* b_i and a *quantity* q_i . The semantics are that agent i is supplying a quantity q_i of the security and an amount b_i of money to be traded in the market. For simplicity, we assume that there are no restrictions on credit or short sales, and so an agent's trade is not constrained by her possessions. The market clears in each round by settling at a single price that balances the trade in that round: The clearing price is $p = \sum_i b_i / \sum_i q_i$. At the end of the round, agent i holds a quantity q'_i proportional to the money she bid: $q'_i = b_i/p$. In addition, she is left with an amount of money b'_i that reflects her net trade at price p : $b'_i = b_i - p(q'_i - q_i) = pq_i$. Note that agent i 's net trade in the security is a purchase if $p < b_i/q_i$ and a sale if $p > b_i/q_i$.

After each round, the clearing price p is publicly revealed. Agents then revise their beliefs according to any information garnered from the new price. The next round proceeds as the previous. The process continues until an equilibrium is reached, meaning that prices and bids do not change from one round to the next.

Here, we make a further simplifying restriction on the trading in each round: We assume that $q_i = 1$ for each agent i . This serves two analytical functions: First, it forces trade to occur. Our model has only rational, risk-neutral, informed traders, and the classic no-trade results would apply. As we have seen, there are several reasons why rational traders would want to trade in practice (subsidies, insurance traders, *etc.*). This forced trade assumption allows us to capture this practical fact without the complications of explicitly modeling these reasons. Second, the fact that agents know the volume of other agents' trades improves their ability to learn from prices. This perhaps gives our agents too much power; but as we shall see, there are still situations in which the market does not converge to the correct value.

1.5.3 Agent Behavior

We assume that agents are risk-neutral, myopic,[†] and bid truthfully: Each agent in each round bids his or her current valuation of the security, which is that agent's estimation of the expected payoff of the security. Expectations are computed according to each agent's probability distribution. We assume that there is a common prior probability distribution \mathcal{P} over values of \mathbf{x} shared by all agents; the agents use their private information and the observed prices to update their beliefs via Bayes' rule. We also assume that it is common knowledge that all the agents behave in the specified manner.

[†] Myopic behavior means that agents treat each round as if it were the final round: They do *not* reason about how their bids may affect the bids of other agents in future rounds.

Example 1.8 Consider a market with two agents, who have private bits x_1 and x_2 respectively. Further, assume the prior probability distribution is uniform, so that each of the four possible values for \mathbf{x} will have a prior probability of $\frac{1}{4}$. Now, we introduce a security F based on the OR function $f(\mathbf{x}) = x_1 \vee x_2$; that is, F eventually pays \$1 if $f(\mathbf{x})$ is 1. Suppose agent 1 observed $x_1 = 0$. Then, conditioned on this information, agent 1 believes $P((x_1, x_2) = (0, 0)) = P((x_1, x_2) = (0, 1)) = \frac{1}{2}$. Then agent 1's initial expectation of the value of F is 0.5; hence, in our model, she would bid $b_1 = 0.5$ in the first round of trading. On the other hand, suppose that agent 2 observed $x_2 = 1$. Then, her posterior beliefs would be $P((x_1, x_2) = (0, 1)) = P((x_1, x_2) = (1, 1)) = \frac{1}{2}$. She would know for certain that f is 1, and would bid $b_2 = 1$. The clearing price of the market after the first round would thus be 0.75.

1.5.4 Equilibrium Price Characterization

We now turn to analyzing the equilibrium trading price in the market. Our analysis builds on powerful results from the economic literature on common knowledge of aggregates.

Recall that there is a set of possible states Ω , together with a common prior probability distribution \mathcal{P} . As trading proceeds, some possible states can be logically ruled out, but the relative likelihoods among the remaining states are fully determined by the prior \mathcal{P} . So the common knowledge after any stage is completely described by the set of states that an external observer—with no information beyond the sequence of prices observed—considers possible (along with the prior). Similarly, the knowledge of agent i at any point is also completely described by the set of states she considers possible. We use the notation S^r to denote the common-knowledge possibility set after round r , and S_i^r to denote the set of states that agent i considers possible after round r .

Initially, the set of states considered possible by an external observer is the set $S^0 = \Omega$. However, each agent i also knows the value of her bit x_i ; thus, her knowledge set S_i^0 is the set $\{\mathbf{y} \in \Omega \mid y_i = x_i\}$. Agent i 's first-round bid is her conditional expectation of the event $f(\mathbf{x}) = 1$ given that $\mathbf{x} \in S_i^0$. All the agents' bids are processed, and the clearing price p^1 is announced. From his knowledge of the prior and the information structure, the external observer can determine the function $price^1(\mathbf{x})$ that relates the first round price to the true state \mathbf{x} . Thus, he can *rule out any vector \mathbf{x} that would have resulted in a different clearing price.*

Thus, the common knowledge after round 1 is the set $S^1 = \{\mathbf{y} \in S^0 \mid price^1(\mathbf{y}) =$

p^1 }. Agent i knows the common knowledge and, in addition, knows the value of bit x_i . Hence, after every round r , the knowledge of agent i is given by $S_i^r = \{y \in S^r | y_i = x_i\}$. Note that, because knowledge can only improve over time, we must always have $S_i^r \subseteq S_i^{r-1}$ and $S^r \subseteq S^{r-1}$. Thus, after a finite number of rounds, we must reach an equilibrium after which no player learns any further information. We use S^∞ to denote the common knowledge at this point, and S_i^∞ to denote agent i 's knowledge at this point. Let p^∞ denote the clearing price at equilibrium.

We now state (without proof) a result that follows immediately from known results on common knowledge of aggregates:

Theorem 1.9 *In the Boolean market, the following conditions must hold at equilibrium:*

$$P(f(\mathbf{y}) = 1 | \mathbf{y} \in S^\infty) = p^\infty \quad (1.1)$$

$$\forall i \ P(f(\mathbf{y}) = 1 | \mathbf{y} \in S_i^\infty) = p^\infty \quad (1.2)$$

Informally, Theorem 1.9 tells us that, at equilibrium, all agents must have exactly the same expectation of the value of the security, and that this must agree with the expectation of an uninformed observer. Note that they may still have differing knowledge sets, as long as conditioning on their respective knowledge sets yields the same expectation. However, reaching agreement is not sufficient for the purposes of information aggregation. We also want the price to reveal the actual value of $f(\mathbf{x})$. The following example shows that it is possible that the equilibrium price p^∞ of the security F will not be either 0 or 1, and so we cannot infer the value of $f(\mathbf{x})$ from it.

Example 1.10 Consider two agents 1 and 2 with private input bits x_1 and x_2 respectively. Suppose the prior probability distribution is uniform, *i.e.*, $\mathbf{x} = (x_1, x_2)$ takes the values $(0, 0)$, $(0, 1)$, $(1, 0)$, and $(1, 1)$ each with probability $\frac{1}{4}$. Now, suppose the aggregate function we want to compute is the *XOR* function, $f(\mathbf{x}) = x_1 \oplus x_2$. To this end, we design a market to trade in a Boolean security F , which will eventually payoff \$1 iff $x_1 \oplus x_2 = 1$.

If agent 1 observes $x_1 = 1$, she estimates the expected value of F to be the probability that $x_2 = 0$ (given $x_1 = 1$), which is $\frac{1}{2}$. If she observes $x_1 = 0$, her expectation is the conditional probability that $x_2 = 1$, which is also $\frac{1}{2}$. Thus, in either case, agent 1 will bid 0.5 for F in the first round. Similarly, agent 2 will also always bid 0.5 in the first round. Hence, the first round of trading ends with a clearing price of 0.5. From this, agent 2 can infer that agent 1 bid 0.5, but this gives her no information about the value of x_1 —it is still equally likely to be 0 or 1. Agent 1 also gains no information from the first

round of trading, and hence neither agent changes her bid in the following rounds. Thus, the market reaches equilibrium at this point. As predicted by Theorem 1.9, both agents have the same conditional expectation (0.5) at equilibrium. However, the equilibrium price of the security F does not reveal the value of $f(x_1, x_2)$, even though the combination of agents' information is enough to determine it precisely.

1.5.5 Characterizing Computable Aggregates

We now give a necessary and sufficient characterization of the class of functions f such that, for any prior distribution on \mathbf{x} , the equilibrium price of F will reveal the true value of f . We show that this is exactly the class of *weighted threshold functions*:

Definition 1.11 A function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ is a **weighted threshold function** iff there are real constants $w_0, w_1, w_2, \dots, w_n$ such that

$$f(\mathbf{x}) = 1 \text{ iff } w_0 + \sum_{i=1}^n w_i x_i \geq 1$$

We now state the following results; due to space restrictions, we do not include the proof. The OR and XOR examples (Examples 1.8 and 1.10) give some insight into these results.

Theorem 1.12 *If f is a weighted threshold function, then, for any prior probability distribution \mathcal{P} , the equilibrium price of F is equal to $f(\mathbf{x})$.*

Theorem 1.13 *Suppose $f : \{0, 1\}^n \rightarrow \{0, 1\}$ cannot be expressed as a weighted threshold function. Then there exists a prior distribution \mathcal{P} for which the price of the security F does not converge to the value of $f(\mathbf{x})$.*

1.5.6 Convergence Time

The model also enables analysis of the number of rounds it takes for the market to converge. We state (but do not prove) the results here:

Theorem 1.14 *Let f be a weighted threshold function with n inputs, and let \mathcal{P} be an arbitrary prior probability distribution. Then, after at most n rounds of trading, the price reaches its equilibrium value $p^\infty = f(\mathbf{x})$.*

Theorem 1.15 *There is a function \mathcal{C}_n with $2n$ inputs and a prior distribution \mathcal{P}_n such that, in the worst case, the market takes n rounds to reveal the value of $\mathcal{C}_n(\cdot)$.*

1.6 Open Questions

We conclude with some open questions and future work.

Combinatorial Prediction Markets

- Section 1.3 discusses combinatorial prediction markets from the auctioneer’s perspective. The bidder’s perspective is also interesting to examine. How should bidders choose boolean formulas ϕ , perhaps subject to constraints or penalties on the number or complexity of bids? How should bidders choose quantities and prices?
- Are there less expressive bidding languages that admit polynomial matching algorithms yet are still practically useful and interesting?
- Although exact matching in general is intractable, are there good heuristics that achieve matches in many cases, or approximate a matching? In particular, is there a practically useful logical reduction algorithm for finding matches?
- We can study permutation combinatorics instead of Boolean combinatorics. In this case, the state space Ω consists of all possible orderings of a set of statistics, say finish times in a horse race. Then a high-level bidding language might allow wagers on events like “ X_1 will win”, “ X_1 will finish in the top 3”, “ X_1 will beat X_2 ”, etc. Are there natural bidding languages with tractable matching problems in this setting?
- Can the auctioneer share the surplus partially or fully with the traders? What are the incentive properties of the resulting mechanisms?
- What is the complexity of finding a match between a single new order and a set of old orders known to have no matches among them? The objective function would be to satisfy as much of the new order as possible, giving the advantage of any price differences to the new order. (This is the standard continuous double auction rule.)
- We may consider a market to be in *computational equilibrium* if no computationally-bounded player can find a strategy that increases utility. Can a market achieve a computational equilibrium that is not a true equilibrium? Under what circumstances?

Automated Market Makers

- For every bidding language that admits a polynomial time matching algorithm as defined in Section 1.3, does there exist a corresponding polynomial time market scoring rule market maker algorithm?

- The market makers of Section 1.4 can be considered as simple *online algorithms* (see Chapter 16). Orders arrive one at a time and the market maker must decide to (partially) accept or reject the order under a constraint of bounded worst-case loss. Are there other online algorithms that can accept more orders for the same worst-case bound on loss?

Distributed Computation through Markets

- The market model in Section 1.5 assumes that the clearing price is known with unlimited precision. Further, the model assumes that none of the traders are misinformed or irrational. What aggregates can be computed even in the presence of noisy prices and traders?
- If the agents have computed the value of the function and a small number of input bits are switched, can the new value of the function be computed incrementally and quickly?
- In the model presented, distributed information is aggregated through a centralized market computation. Can we find a good distributed-computational model of a decentralized market?
- What is the complexity of the computations that agents must do to update their beliefs after each round?
- The model in Section 1.5 directly assumes that agents bid truthfully. Is there a tractable model that assumes only rationality and solves for the resulting game-theoretic solution strategy?
- The negative results in Section 1.5 (Theorems 1.13 and 1.15) examine worst-case scenarios, and thus involve very specific prior probability distributions and initial information states. On the other hand, simulations seem to suggest that almost every threshold function’s expected convergence time is near constant, where expectation is taken over the common prior. Can we prove results about average-case convergence?
- Non-threshold functions can be implemented by combining two or more threshold functions. What is the minimum number of threshold securities required to implement a given function? Are there ways to configure securities to speed up convergence to equilibrium?

1.7 Exercises

- Describe how the market scoring rule market maker of Section 1.5 can be extended to handle limit orders of the form “buy at most q units of S_ϕ at price less than or equal to p ”. For simplicity, assume that partially filled limit orders do not remain active in the system.

- A straightforward implementation of a combinatorial market maker, where $\Omega = 2^{|\mathcal{E}|}$, requires exponential space to explicitly maintain the vector \vec{q} , the number of shares outstanding of each of the $2^{|\mathcal{E}|}$ possible outcomes (states). Derive a polynomial-space version of a combinatorial logarithmic market scoring rule market maker, where the input is the list of previously accepted orders and the new order and the output is $C(\vec{q})$. Orders can either be combined orders or compound orders, as defined in Section 1.3.
- Define the *conditional cost function* for the logarithmic market scoring rule as $C_\psi(\vec{q}) = b \ln(\sum_{j:\omega_j \in \psi} e^{q_j/b})$: the same cost function as before but summed only over states in ψ . The conditional cost function can be used to price conditional securities. The cost to buy δ shares of $S_{\phi|\psi}$ is $C_\psi(\vec{q} + \delta \cdot 1_\phi) - C_\psi(\vec{q})$. Also, by Bayes's Rule, we know that the instantaneous price of $S_{\phi|\psi}$ equals the price of $S_{\phi \wedge \psi}$ divided by the price of S_ψ .
 - (i) Verify that the price of $S_{\phi|\psi}$ defined in this way integrated from 0 to δ equals $C_\psi(\vec{q} + \delta \cdot 1_\phi) - C_\psi(\vec{q})$.
 - (ii) After a trader purchases δ shares of $S_{\phi|\psi}$, what is the new quantity vector \vec{q}_{new} ? (Hint: it is *not* $\vec{q}_{old} + \delta \cdot 1_\phi$.)
- Consider the two-agent “OR” market of Example 1.8. Suppose that $X_1 = 0$ and $X_2 = 1$. Prove that bidding truthfully is not a Nash equilibrium. To do so, it suffices to show that if bidder 1 bids truthfully, then bidder 2's optimal bid is not truthful.

Acknowledgments

We thank Yiling Chen for help and contributions. We thank Joan Feigenbaum, Lance Fortnow, Joe Kilian, and Michael Wellman.

1.8 Bibliographic Notes

This section surveys some of the most directly relevant related work; a more extensive bibliography will be made available on the authors' home pages. We also point readers to excellent survey articles on prediction markets by Tziralis and Tatsiopoulos [TT06], Wolfers and Zitzewitz [WZ04, WZng], and Berg and Rietz [BR03].

A number of studies investigate forecast accuracy and trader behavior on the Iowa Electronic Market, one of the longest-running active prediction markets. Berg *et al* [BFNR01] surveys this work. Other empirical studies examine markets on TradeSports.com, an Irish betting exchange [WZ06, WSZ07, Tet04, Tet06]. Perhaps surprisingly, even play-money market games perform well compared to experts and real-money markets [CCMP05, PLGN01,

PLNG01, SSWPG04, SS03, MDF⁺05]. The field tests at Hewlett Packard were conducted by Chen and Plott [CP02, Pl00]. Sunder [Sun95] reviews a number of laboratory experiments involving prediction markets.

A common concern is that prediction market prices may be manipulated by wealthy traders with ulterior motives. Rhode and Strumpf [RS06] analyze manipulation attempts in real markets and find that the effects of manipulations are typically minimal and short lived. Hanson *et al* [HOP06] find that markets appear robust to manipulation in a laboratory setting.

The theory of rational expectations was introduced by Muth [Mut61] and further developed by Lucas [Luc72]. The article by Grossman [Gro81] is a good introductory survey. No-trade theorems [MS82] have their roots in the theory of common knowledge [Aum76]. Several authors discuss a procedural explanation of rational expectations, showing that repeated announcement of an aggregate statistic of the agents' beliefs will drive the agents to a consensus, if they begin with common priors [Han98, MP86, MP90, NBG⁺90]. The oft-cited *efficient market hypothesis* [Fam70] is rooted in rational expectations theory.

The analysis of combinatorial prediction markets in Section 1.3 follows Fortnow *et al* [FKPW05]. Chen *et al* [CFNP07] conduct an analogous study of permutation combinatorics. Bossaerts *et al* [BFL02] introduce the *combined value trading* framework, providing algorithms for clearing prediction markets when combined orders are allowed.

The description in Section 1.3.2 of compact prediction markets that take advantage of (conditional) independence among events is based on work by Pennock and Wellman [PW00, PW05].

Market scoring rules were introduced by Hanson [Han03, Han06]. Hanson describes how the market scoring rule market maker is especially well suited for combinatorial prediction markets, and discusses some of the associated computational challenges. Scoring rules have long been used to measure forecast accuracy [Sav71, WM68]. Dynamic parimutuel markets were introduced by Pennock [Pen04].

Section 1.5 follows the work of Feigenbaum *et al* [FFPS05]. Chen *et al* [CMC06] examine an extended model where aggregate uncertainty remains in equilibrium. Theorem 1.9 follows from a result due to McKelvey and Page; see Nielsen *et al* [NBG⁺90] for more details. The market model is based on a model due to Shapley and Shubik [SS77]. Ronen and Wahrman [RW05] investigate a slightly different model of *prediction games*, in which a mechanism designer seeks to compute a function of agents' information, but agents incur a cost to access their own information.

Bibliography

- [Aum76] Robert Aumann. Agreeing to disagree. *Annals of Statistics*, 4:1236–1239, 1976.
- [BFL02] Peter Bossaerts, Leslie Fine, and John Ledyard. Inducing liquidity in thin financial markets through combined-value trading mechanisms. *European Economic Review*, 46:1671–1695, 2002.
- [BFNR01] J. E. Berg, R. Forsythe, F. D. Nelson, and T. A. Rietz. Results from a dozen years of election futures markets research. In C. A. Plott and V. Smith, editors, *Handbook of Experimental Economic Results (forthcoming)*. 2001.
- [BR03] J. E. Berg and T. A. Rietz. Prediction markets as decision support systems. *Information Systems Frontier*, 5:79–93, 2003.
- [CCMP05] Y. Chen, C. H. Chu, T. Mullen, and D. M. Pennock. Information markets vs. opinion pools: An empirical comparison. In *Proceedings of the Sixth ACM Conference on Electronic Commerce (EC’05)*, Vancouver, Canada, June 2005.
- [CFNP07] Yiling Chen, Lance Fortnow, Evdokia Nikolova, and David M. Pennock. Betting on permutations. Working Paper, 2007.
- [CMC06] Yiling Chen, Tracy Mullen, and Chao-Hsien Chu. An in-depth analysis of information markets with aggregate uncertainty. *Electronic Commerce Research*, 6(2):201–221, 2006.
- [CP02] K. Y. Chen and C. R. Plott. Information aggregation mechanisms: Concept, design and implementation for a sales forecasting problem. Working paper No. 1131, California Institute of Technology, Division of the Humanities and Social Sciences, 2002.
- [Fam70] E. F. Fama. Efficient capital market: A review of theory and empirical work. *Journal of Finance*, 25:383–417, 1970.
- [FFPS05] Joan Feigenbaum, Lance Fortnow, David M. Pennock, and Rahul Sami. Computation in a distributed information market. *Theoretical Computer Science*, 343:114–132, 2005. (A preliminary version appeared in the 2003 ACM Conference on Electronic Commerce).
- [FKPW05] Lance Fortnow, Joe Kilian, David M. Pennock, and Michael P. Wellman. Betting boolean-style: A framework for trading in securities based on logical formulas. *Decision Support Systems*, 39(1):87–104, 2005.
- [Gro81] Sanford J. Grossman. An introduction to the theory of rational expectations under asymmetric information. *Review of Economic Studies*, 48(4):541–559, 1981.
- [Han98] Robin Hanson. Consensus by identifying extremists. *Theory and Decision*, 44(3):293–301, 1998.
- [Han03] Robin Hanson. Shall we vote on values, but bet on beliefs? Working Paper, 2003.
- [Han06] Robin Hanson. Logarithmic market scoring rules for modular combinatorial information aggregation. *Journal of Prediction Markets*, 1(1), 2006.
- [HOP06] Robin Hanson, Ryan Oprea, and Dave Porter. Information aggregation and manipulation in an experimental market. *Journal of Economic Behavior and Organization*, page (to appear), 2006.
- [Luc72] Robert E. Lucas. Expectations and the neutrality of money. *Journal of Economic Theory*, 4(2):103–24, 1972.
- [MDF⁺05] Bernard Mangold, Mike Dooley, Gary W. Flake, Havi Hoffman, Tejaswi Kasturi, David M. Pennock, and Rael Dornfest. The tech buzz game. *IEEE Computer*, 38(7):94–97, July 2005.

- [MP86] Richard McKelvey and Talbot Page. Common knowledge, consensus, and aggregate information. *Econometrica*, 54(1):109–127, January 1986.
- [MP90] Richard D. McKelvey and Talbot Page. Public and private information: An experimental study of information pooling. *Econometrica*, 58(6):1321–1339, 1990.
- [MS82] Paul Milgrom and Nancy Stokey. Information, trade, and common knowledge. *Journal of Economic Theory*, 26:17–27, 1982.
- [Mut61] John A. Muth. Rational expectations and the theory of price movements. *Econometrica*, 29(6):315–335, 1961.
- [NBG⁺90] Lars Tyge Nielsen, Adam Brandenburger, John Geanakoplos, Richard McKelvey, and Talbot Page. Common knowledge of an aggregate of expectations. *Econometrica*, 58(5):1235–1238, 1990.
- [Pen04] David Pennock. A dynamic parimutuel market for information aggregation. In *Proceedings of the Fourth Annual ACM Conference on Electronic Commerce (EC '04)*, June 2004.
- [PLGN01] David M. Pennock, Steve Lawrence, C. Lee Giles, and Finn Årup Nielsen. The real power of artificial markets. *Science*, 291:987–988, February 9 2001.
- [PLNG01] David M. Pennock, Steve Lawrence, Finn Årup Nielsen, and C. Lee Giles. Extracting collective probabilistic forecasts from web games. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 174–183, 2001.
- [Plo00] Charles R. Plott. Markets as information gathering tools. *Southern Economic Journal*, 67(1):1–15, 2000.
- [PW00] David M. Pennock and Michael P. Wellman. Compact securities markets for Pareto optimal reallocation of risk. In *Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI-2000)*, pages 481–488, July 2000.
- [PW05] David M. Pennock and Michael P. Wellman. Graphical models for groups: Belief aggregation and risk sharing. *Decision Analysis*, 2(3):148–164, 2005.
- [RS06] Paul W. Rhode and Koleman S. Strumpf. Manipulating political stock markets: A field experiment and a century of observational data. Working Paper, 2006.
- [RW05] Amir Ronen and Lyron Wharmann. Prediction games. In *Workshop on Internet and Network Economics*, 2005.
- [Sav71] Leonard Savage. Elicitation of personal probabilities and expectations. *Journal of the American Statistical Association*, 66(336):783–801, 1971.
- [SS77] Lloyd Shapley and Martin Shubik. Trade using one commodity as a means of payment. *Journal of Political Economy*, 85:937–968, 1977.
- [SS03] M. Spann and B. Skiera. Internet-based virtual stock markets for business forecasting. *Management Science*, 49(10):1310–1326, 2003.
- [SSWPG04] E. Servan-Schreiber, J. Wolfers, D. M. Pennock, and B. Galebach. Prediction markets: Does money matter? *Electronic Markets*, 14(3):243–251, 2004.
- [Sun95] S. Sunder. Experimental asset markets. In J. H. Kagel and A. E. Roth, editors, *The Handbook of Experimental Economics*, pages 445–500. Princeton University Press, 1995.
- [Tet04] Paul C. Tetlock. How efficient are information markets? evidence from an online exchange. Working Paper, 2004.
- [Tet06] Paul C. Tetlock. Does liquidity affect securities market efficiency? Working Paper, 2006.
- [TT06] George Tziralis and Ilias Tatsiopoulos. Prediction markets; An extended

- literature review. *Journal of Prediction Markets*, 1(1), 2006.
- [WM68] R. L. Winkler and A. H. Murphy. Good probability assessors. *Journal of Applied Meteorology*, 7:751–758, 1968.
- [WSZ07] Justin Wolfers, Erik Snowberg, and Eric Zitzewitz. Partisan impacts on the economy: Evidence from prediction markets and close elections. *Quarterly Journal of Economics*, 122(2), 2007.
- [WZ04] Justin Wolfers and Eric Zitzewitz. Prediction markets. *Journal of Economic Perspective*, 18(2):107–126, 2004.
- [WZ06] Justin Wolfers and Eric Zitzewitz. Using markets to inform policy: The case of the iraq war. Under Review, 2006.
- [WZng] Justin Wolfers and Eric Zitzewitz. Prediction markets in theory and practice. In Larry Blume and Steve Durlauf, editors, *The New Palgrave Dictionary of Economics*, 2nd edition. Forthcoming.