A Decentralised Peer-Prediction Market

CS907 Dissertation Project - Dissertation

Thomas Archbold
1602581

Department of Computer Science
University of Warwick

August 30, 2020

Supervised by Matthias Englert

Contents

1 Introduction

Prediction markets are exchange-traded markets[1] which allow users to trade on the outcomes of future events as opposed to traditional financial instruments. Users participate by placing bets and buying or selling shares in the markets these bets give rise to. Since users stake their own money, market prices should indicate the true beliefs of the userbase and the perceived likelihood the events have of occurring. Different users will have different beliefs and knowledge informing their decisions, and prediction markets provide a means of aggregating information on events of interest using "the wisdom of the crowd". Shares in these markets are usually traded between \$0 and \$1, and for binary events a market will typically pay out \$1 for every share held for a positive outcome, and \$0 otherwise.

Traditional prediction markets are centralised in the sense that the system provides the bets for the users to trade in, and traders simply choose the price point and size of stake at which to participate. Since the system knows all possible bets beforehand it is simple to allow it to determine the outcomes of the markets and pay out the winnings to stakeholders accordingly. There are two issues with this centralised approach: firstly, it restricts the types of bets that can be made, since they must be explicitly offered by the market maker; secondly, it operates on trust – there is nothing to stop the central market maker from manipulating the system for their own gain. We aim for a system that avoids both of these issues.

In this project we implement a *decentralised* prediction market in which it is up to the userbase itself to define the markets and determine the outcomes of these events – these outcomes are decided upon by consensus among a group of users, known as arbiters. This removes the need for a trusted centre, however with no central moderator bets may become ambiguous or their outcomes subjective, and arbiters may still attempt to manipulate the outcome of the market for their own gain by submitting false reports. It is also important that users continue to act according to their true beliefs so we may learn about the true public sentiment on the events. We base our design on the incentive compatible outcome determination mechanism proposed by Freeman, Lahaie, and Pennock [10], which allows us to crowdsource market outcomes while incentivising users to act truthfully in all stages of the prediction market.

The rest of this dissertation is structured as follows: in Section 2 we give an overview of the current literature on algorithmic mechanism design and prediction markets, and discuss recent implementations of decentralised markets. In Section 3 we outline the motivations for undertaking the project and why we believe it to be worthwhile, and detail the successes that have been achieved so far. Section 4 covers the high-level design of the market and introduces the theoretical model upon which our implementation is based, as well as issues related to project management and ethical considerations. In Section **??** we then discuss our implementation of the prediction market, including the tools we have used to do so. Finally in Section **??** we reflect on the project's successes and failures and suggest areas for further development.

2 Background

2.1 Setup

Consider the following problem within algorithmic mechanism design known as the *information aggregation problem* [16, Ch. 26]. An individual known as the "aggregator" wishes to obtain a prediction about an uncertain variable which will be realised at some point in the future. There are a number of individuals known as the "informants" who each hold sets of information about the variable's outcome. The goal is to design a mechanism that extracts the relevant information from the informants and uses it to provide a prediction of the variable's realisation. In an ideal setting the mechanism should produce the same prediction as an omniscient forecast that has access to all information available to all informants. This may not be viable in practice, since each agent's information is private information, and so the mechanism must incentivise them to act in the desired truth-telling manner.

A prediction market is one mechanism that can provide such a forecast. In this setting the aggregator creates a financial security whose payoff is tied to the outcome of the variable. In the simpler case of binary events, such a security may pay out \$1 for each share held if the variable has a "true" or "yes" outcome, and \$0 otherwise, however markets can be created for other types including discrete ("will the result of the match be a home win, away win, or a draw?"), continuous ("what will be the highest measured temperature in Coventry in September?"), or any combination of these types. Informants

[1] Markets in which all transactions are routed through a central source.

are then able to participate in the market induced by the security by trading shares according to their beliefs: those who believe, for example, that global warming is real might buy shares at a given price in the market, "the global average temperature in 2020 will be higher than that in 2019", while deniers may be inclined to sell shares. The share price will be adjusted accordingly, and the aggregator can view the current price as the informant's combined belief of the outcome of the event.

In this model there is a set Ω of possible states of the world and at any moment the world is in exactly one state $\omega \in \Omega$, though the informants do not know which. Each informant i may however possess partial information regarding this true state, and this is represented by a partition π_i of Ω. The agent knows in which subset of this partition the true world state lies, but does not know the exact member of which is true. Given n agents, their combined information $\hat{\pi}$ is the coarsest common refinement of the partitions $\pi_1, \ldots, \pi_n$.[2]

We also assume a common prior probability distribution $P \in \Delta^{\Omega}$ which describes the probabilities that all agents assign to the different world states before receiving any information. Once each agent receives their partial information, they form their posterior beliefs by restricting the common prior to the subset of their partition in which they know the true state to lie. For our purposes, a *forecast* is an estimate of the expected value of the function $f : \Omega \to \{0, 1\}$, known as an *event*, which equals one for exactly one subset of Ω and 0 otherwise.

As mentioned, there is an ideal "omniscient" forecast that uses the distribution P restricted to the subset of $\hat{\pi}$ in which the true world state lies but which is impractical given the private nature of each agent's information. The goal is therefore a mechanism to incentivise agents to reveal their private information such that in equilibrium we achieve a forecast as close as possible to the omniscient one. Prediction markets offer the agents the chance of financial gain for revealing information regarding the expected value of $f(\omega)$, and the share price of the security can be interpreted as the collective forecast of the agents. In Section 4 we shall outline the different approaches one can take to designing the prediction market mechanism itself and the one among them, a Market Scoring Rule, that we implement. Importantly, Market Scoring Rules are one of the ways to increase liquidity in the market as the system itself assumes the opposite side to any trade, meaning users may participate even when no other user wishes to buy or sell for what they are asking or bidding. This means a trade is always able to be executed, although without care the system could make consistent losses.

2.2 Literature Review

The Iowa Electronic Markets (IEM) are real-money prediction markets developed by the University of Iowa [3] that have been running since 1988. They allow users to buy and sell contracts based on the outcome of U.S. political elections and economic indicators, and are currently offering markets for the winning party of the 2020 U.S. presidential election, the vote share between the Democratic and Republican parties in the 2020 U.S. presidential election, and the compositions of the houses of Congress, House of Representatives, and U.S. Senate after the outcome of the 2020 U.S. congressional elections. The number of markets offered is small and the topics are kept relevant to current events, meaning there is likely to be high liquidity for any security a user wishes to trade in. This has allowed the markets to predict the results of political elections with more accuracy and less error than traditional polls: for the presidential elections between 1988 and 2000, three-quarters of the time the IEM's market price on the day each poll was released was more accurate for predicting vote share than the poll itself [18, pg. 19]. These markets inspired similar markets in the forms of the Hollywood Stock Exchange, NewsFutures, and the Foresight Exchange report, which achieved similar successes despite not using real money.

An issue with these markets is that they are restrictive in the bets they offer. Although this can be beneficial in that they provide a focused and liquid market in which to trade, it leaves them potentially less interesting to interact with. A combinatorial prediction market is a solution to this and drastically increases the number of events that can be bet on and outcomes predicted by offering securities on interrelated propositions that can be combined in various ways. One example of such a market is that of *Predictalot* [6], a combinatorial prediction market developed by Yahoo! that allowed users to trade securities in the 2010 NCAA Men's Division I Basketball Tournament. The tournament sees the top 64 teams play 63 games in a knockout competition, yielding a total outcome space of size 2^{63}. *Predictalot* then kept track of the odds, computing them by scanning through all of the predictions made by users. This prediction market was the original inspiration for this project in investigating prediction markets. Using a Market Scoring Rule for such a market would involve computing a summation over

[2] A partition α of a set X is a refinement of a partition ρ of X if every element of α is a subset of some element of ρ. In this case α is *finer* than ρ and ρ is *coarser* than α.

the entire outcome space Ω, an intractable, #P-hard problem akin to counting the number of variable assignments that satisfy a CNF formula, or or the number of subsets in a list of integers that sum to zero. Instead, they employ an implementation of importance sampling, a technique for estimating properties of a particular probability distribution using only samples generated from a different distribution. This "naive" approach is then improved upon by the work of Dudík, Lahaie, and Pennock [9], who use convex optimisation and constraint generation to develop a market maker which is tractable. This approach lies somewhere between treating all securities as independent and a fully combinatorial, "ideal" market maker, propagating information among related securities for more complete information aggregation. The ways in which their odds are calculated are also natural: for example, a large bet on a team to win the entire tournament automatically increases the odds that the same team will progress past the first round, since they would not be able to win the competition without doing so. This work is then improved upon by Kroer, Dudík, Lahaie, and Balakrishnan [14], in which they use integer programming to achieve arbitrage-free trades, or always profitable risk-free trades. On top of achieving bounded loss, a crucial element behind a market mechanism operating in the real world and avoiding bankruptcy, avoiding arbitrage is desirable as it leads to more accurate forecasts: since users cannot make risk-free profits, they are forced to bet according to their true beliefs.

All examples so far have involved a centralised market mechanism. These types of systems involve a central authority providing the bets upon which users may bet and then verifying their outcome. *Decentralised* markets allow the users themselves to define their own bets and trade shares in them. These types of systems involve a central authority providing the bets upon which users may bet and then verifying their outcome. These types of markets allow the users themselves to define their own markets by providing custom bets and them trading shares in them. Several examples of decentralised markets exist and they are often implemented with cryptocurrencies. Peterson et al. [17] study the setting and use it to implement the oracle at the heart of *Augur* [1], a decentralised prediction market built upon the Ethereum blockchain that launched in 2018. It allows users to offer predictions on any topic, and markets may be either categorical, which are similar to binary markets in which the winner takes all, or scalar, which offer users a spectrum of outcomes in which to invest. For example, users may bet that the global average temperature for 2020 will lie in a certain range. As in many decentralised markets, outcomes of events are then resolved by the users, and in the case of *Augur* users are incentivised to report truthfully by way of paying reporting fees. This amounts to users depositing tokens to back their report, and token holders are then entitled to the trading fees generated. Although this persuades against manipulation, it has not been shown whether this system achieves any theoretical guarantees of truthful reporting.

As can often be the case with real-money markets, the platform had quickly devolved into an assassination market [8] – originally this referred to the case where users created markets on the deaths of certain people, which then incentivised their assassination. A user could stand to profit by placing a bet on the exact time of their death, and ensure this bet was profitable by assassinating the subject. More generally this refers to the users of a prediction market having the ability to influence a market's outcome and acting on this opportunity. Another issue with *Augur* is the option to report a market's outcome as "invalid": this is for the case where the user-made bet is too ambiguous to decided, such as, "Bayern Munich will play well against Paris Saint Germain".

Other decentralised markets based on cryptocurrencies exist, including *Omen* [5] and *Hivemind* [2]. The former is similar to *Augur* in that it allows users to create markets for any bet they like and whose outcomes are not decided by the system itself. Whereas *Augur* uses a reputation system in the form of requiring users to back their report of a market's outcome with \$REP tokens, *Omen* asks the market creator to supply an "oracle" through which the outcome can be determined. They note that this oracle can even be *Augur*. Although this may solve the "invalid" outcome option for ambiguous bets, it may introduce bias into the process of outcome determination. For example, suppose a user creates a market for "The Democratic nominee will tell a lie during tonight's debate" and lists the oracle as the conservative news channel, Fox News. Users would then trade on how they think the oracle will report the outcome, and not what they believe the outcome will be themselves. An important aspect of decentralised markets must therefore be that the outcome is determined by the community, not a single source.

In contrast to *Augur*, which implements a traditional orderbook using Ethereum, *Omen* uses an automated market maker to provide liquidity to its securities. Two approaches to this include using a Market Scoring Rule to update the odds on a given event, and implementing a parimutuel market where users compete for a share of the total money wagered while the share price varies dynamically according to some cost function. Hanson [11] shows that we can use any strictly proper scoring rule to

implement an automated market maker: with such scoring rules, agents maximise their expected utility by truthfully revealing their predictions. In particular, in this project we implement the peer prediction market introduced by Freeman, Lahaie, and Pennock [10], which specifies a market that uses a Market Scoring Rule to trade bets and crowdsources outcome determination, similarly to *Augur*, by asking users for reports. All they require is that the rule is strictly proper, giving plenty of choice to study the effects different rules have on user behaviour. While the choice of scoring rule is less important than the mechanism by which market outcomes are determined, a recent work by Liu, Wang, and Chen [15] introduces scoring rules for the setting where the aggregator has access only to user reports, which they call Surrogate Scoring Rules (SSRs). This appears to be an interesting avenue to further explore and adapt to a prediction market. One assumption they make, however, seems incompatible with the decentralised setting in that they require all events to be independent. Given that users can create a market for *any* bet, this condition is impossible to ensure. Since SSRs can be strictly proper under certain conditions, they may be applicable as the Market Scoring Rule in [10].

Other prediction markets existed in *PredictIt* [7] and *InTrade* [4], both offering markets for various political and economic events. However, both experienced disputes largely related to the wording of the available to trade and the ambiguity in their resolution. For example, a bet offered on *PredictIt* was, "Who will be Senate-confirmed Secretary of State on March 31, 2018?", and although Rex Tillerson was fired in the middle of March, he was officially the secretary of state until midnight of the 31st, leading to confusion among users in what they are trading on and therefore inaccuracies in the predictions it elicited. This will be a problem inherent to any prediction market that allows users to specify their own bets, and while the work of Freeman et al. [10] evades the problem of determining the outcome by setting it to the proportion of users reporting a "yes" outcome, this does nothing to penalise the initial creator for introducing an ambiguous market.

3 Goals

3.1 Core Features

The goal of this project is to implement a truthful decentralised prediction market in which users may specify the bets on which to trade. The market outcomes will be decided by peer prediction, in which events are settled by a subset of the users known as arbiters. A user may even act as an arbiter in a market in which they themselves may hold a stake. Specifically, we seek to:

- create a web application on which to host the prediction market
- allow users to create custom bets
- implement a trading mechanism that allows users to buy and sell shares in the user-made securities
- crowdsource outcome determination using reports from arbiters who may hold positions in the market
- incentivise truthful behaviour at all stages in the mechanism

These goals will largely be achieved by implementing the mechanism outlined by Freeman et al. [10], albeit wit several practical modifications. The first three goals cover core functionality of any decentralised prediction market, while the fourth is concerned with the tuning of system parameters, before and during execution, in order to ensure that users do not manipulate the mechanism. In this users are not only discouraged from attempting to "game" the system for personal gain, they are hurt for doing so.

Other non-essential features but highly desirable for strong user experience include asynchronous communication with the server in order to display up-to-date pricing information to the user without a page refresh, and the automated closing of markets. Both of these features would make the system straightforward and intuitive to use. Moreover, it would allow the system to run independently, meaning the market's functioning is only influenced by the community, one of the key points of implementing a decentralised market.

As we will discuss in more detail in Section 4, one aspect of the mechanism is the assumption that the system knows the signal error rates when user's receive news about a market's outcome. This is unrealistic in practice, since we cannot hope to know the exact where each user learns about the outcome of the event nor the accuracy of said source's reporting. Hence we also look to implement a

way in which users do not need to be explicitly asked their estimates of signal accuracy, and instead this is calculated based on their past reporting history. This leaves less opportunity to game the system, the entire point of implementing this prediction market mechanism.

3.2 Stretch Features

With more time, there are plenty of additional features that could be implemented to render the system more intuitive and usable. These include the option to create different types of markets, particularly categorical ones since they would function similarly to binary markets but allow multiple related markets to be expressed more succinctly. Furthermore the option to create and sort markets by categories would help users offer their information more readily if they are especially interested in a certain topic, say politics or sport.

A useful feature to implement would be the tracking of price histories for each security. This would enable graphs to be generated so that users could be more informed on how the forecast of an event has changed over time. This would make the decision to participate at a particular price point more interesting on their part since they are not only estimating their belief on the probability that the event will have a positive outcome, but also on how the other users will act, similar to a real stock market.

Finally, an issue with the mechanism of Freeman et al. as it stands is that it does not directly punish users for creating markets on ambiguous bets. Traders may become confused about the wording, leading to different interpretations and hence different users trading on different beliefs – this is not useful for information aggregation. Although the negative effects are somewhat mitigated in that the very same community that trades in the security also decides on its outcome, there is no mechanism in place to specifically encourage clear bets. This is something that could be improved and would be useful in avoiding the market becoming swamped with overly subjective wagers.

3.3 Motivation

As discussed, there are already numerous prediction markets that exist in the literature. The Iowa Electronic Markets are the longest running and arguably most successful, but the options offered to the users on which to trade are too restrictive. This is a similar issue among all centralised markets, including InTrade, PredictIt, and the Hollywood Stock Exchange. While their success can be attributed to their narrow focus on a particular topic, it seems more interesting to be able to aggregate information from a wider variety of themes, sacrificing perhaps some of the predictive accuracy for more widespread forecasts to be made.

Decentralised prediction markets are not a new concept, however current examples in the literature lack in the functionality they offer. In the case of *Omen*, while they allow any user to create a market, they rely on a single oracle to determine the outcome of the event. This leaves a single point of "failure" in the system and leaves it open to being manipulated. For example, listing a biased news source as the oracle could have a significant effect on the event's outcome. Although this is mitigated somewhat by displaying to traders the oracle chosen, this still encourages them to trade on how they believe the oracle will report the market and not necessarily the market itself. The mechanism by which *Augur* determines market outcomes appears to be an improvement over this, in which multiple reporters from the community back their report of the market outcome with $REP tokens, thus implementing a form of reputation system. However, it does not deal with ambiguous bets elegantly, offering the option for a report a market's outcome as "invalid". Given the inevitability of such markets in a decentralised setting, this is a key weakness to *Augur*.

It is therefore well justified to implement the decentralised peer prediction mechanism of Freeman et al. This not only allows users to create markets for any event they see fit, but also crowdsources market outcomes by relying on reports from the community. Thus instead of relying on a single source of information, which leaves it vulnerable to reporting biases particularly for more ambiguous or subjective bets, it gets a more complete picture of how the users themselves, who are the ones interacting with the market, observed the outcome. This can average out the biases present in any one news source. This also seems to be a better method to deal with ambiguity than in *Augur*, since the outcome of the market can be influenced by a reporter's interpretation of a wager. Another issue with *Augur* is that it has not been shown to achieve any theoretical guarantees, which should be a key consideration in an environment in which rational selfish agents are interacting. Instead, the market we implement is proven to be incentive compatible, meaning it it in a user's best interests to report market outcomes truthfully and not attempt

to manipulate it for their own gain. Although the mechanism is not budget balanced, it can be fully subsidised by a trading fee on each transaction, further making it practical and self-sufficient.

4 Design

4.1 Mechanism Overview

As mentioned our design of the prediction market is based on the peer prediction mechanism proposed by Freeman et al. [10]. In this section we will present the main ideas presented in their work and give an overview of the mechanism itself.

We are interested in setting up a prediction market for outcome of a binary random variable $X \in \{0, 1\}$. We will use the terms "market", "stock", and "security" interchangeably throughout to refer to the entity comprising a wager, such as "Arsenal will beat Tottenham", and a deadline – these two pieces of information are all we need to represent the event X. The mechanism is divided into two main stages: the market stage, where users may buy and sell shares in the securities whose deadlines have not yet passed; and the arbitration stage, where a subset of the users report on the outcome of the security and the payout price per share is computed. In a traditional prediction market on binary events, if the market's outcome was positive then stakeholders with long positions will then be paid out \$1 for each share they own while those with short positions will buy back their shares at a price of \$1 per share. Similarly, if the outcome was negative, long users will have lost money since they receive no money back from their initial investment, while short users will profit as they must "buy back" their shares at a price of \$0 per share. This market is different in that market outcomes are set as the proportion of users that reported a positive outcome. Therefore, even if a user has gone long on a security, this must have been at or below the right price to make a profit, since it is not necessarily the case that 100% of the users reporting on the market agree on its outcome.

Since we rely heavily on user participation for the mechanism to run correctly, it is important that users act in the desired manner. This mechanism incentivises users to act truthfully in two aspects: firstly, users are encouraged to trade on their belief of the market's realised outcome, rather than, for example, how a specific news source will report on it, since this outcome is determined entirely by the community, each individual of which will have access to their own news sources of varying biases; secondly, it is in a user's best interests to report on market outcomes truthfully, since they can receive no better payoff by attempting to manipulate the system. The same is true even if an otherwise rogue reporter held a position in the market. Therefore, we are able to gather accurate public sentiment on the event itself as well as its outcome, and ambiguous securities are dealt with more gracefully.

These considerations also allow us to achieve certain useful guarantees. For example, in order to incentivise reporters to act truthfully, we must pay them more than what they would otherwise gain from attempting to manipulate the system. We can use this knowledge, coupled with what we know stakeholders are expecting to be paid out, in order to bound the amount we must pay to ensure incentive compatibility. In this way we can ensure that the system is sustainable and the system's loss is never unbounded.

In the following sections, we shall outline the mechanism we implement from a theoretical standpoint.

4.1.1 Market Stage

The market stage allows users to create markets for any bet they desire and specifies the how the share price reacts according to user participation. As we implement a decentralised market, we place no restriction on the bet that can be placed (only that it is binary, i.e. it is ultimately either a "yes" or a "no"). In any market, there are two sides to any trade: one side offers some number shares, while there is some party that must be willing to purchase this number of shares at this price. There are a number of options for implementing such a trading mechanism, as we will detail in the following section.

There are a number of options for implementing a market trading mechanism: these include as a continuous double auction, market call auction, or by using an automated market maker. A continuous double auction (CDA) is a method by which buyers are matched with sellers of a stock. The market maker keeps an order book that tracks the bids, submitted by those looking to buy a stock, and the asks, submitted by those looking to sell a stock. For example, suppose Alice says she wishes to buy a share of stock A for \$5, while Bob owns a share of stock \$A and is willing to sell it for \$5. The former is recorded as a bid, and the latter an ask. If the bid an ask for two traders match then the trade is executed: Alice pays Bob \$5 and receive a single share of stock A, and Bob is paid \$5 and gives away his share.

Continuous double auctions are traditionally used in highly liquid markets, where there are many bids for a given ask (and vice versa), such as the New York Stock Exchange. An issue with this approach, however, is that it relies on high liquidity – that is, it requires that there is always a willing buyer and seller for a particular price at a particular quantity of shares. Especially in a prediction market, most prediction markets have far fewer participants than a stock exchange: if Alice is willing to sell a share of stock A for \$6 but no one is willing to buy at that price, then a trade cannot be executed. Such a mechanism relies on other sellers to offer shares at lower prices, and eventually Alice will adjust her ask to the market rate. This is therefore reliant on the market being liquid, which is not always possible in a prediction market, especially when bets are made on niche topics.

To avoid these problems of illiquidity, one can use an automated market maker, in which a price maker is (nearly) always willing to accept both buy and sell orders at a certain price. It is possible that this price changes with the user's interaction with the market.

The market stage allows users to create markets for any event they wish and trade shares in these markets. Since we crowdsource outcome determination, there is no restriction placed on the types of bets users may make other than that their result is a "yes" or a "no". Ambiguous bets are allowed, though likely to suffer in the arbitration stage, since users may have different interpretations on the outcome.

In order to create a market for a user-specified event, we use a *market scoring rule (MSR)*, which is a means of assigning a probability to a set of mutually outcomes. After the wager and deadline have been specified, the MSR simply takes into account the number of shares bought and sold and returns a probability $p \in [0, 1]$. This describes the perceived likelihood amongst users of the event having a positive outcome. We can also use this to set the share price of the security. Using a MSR is different from a traditional market in that there is not a fixed number of shares in circulation: instead, buying into a market increases the total number of shares and selling decreases it. Let q denote the total number of shares of a given security. An agent wishing to buy $q' - q$ shares (i.e. increasing the total number of shares to q') will pay $C(q') - C(q)$, for our choice of convex, differentiable, monotonically increasing scoring rule C. We may tune the behaviour of this cost function by using a liquidity parameter b, such that $C_b(q) := b \cdot C(q/b)$, which controls the responsiveness of C. A lower value of b means that the share price changes more quickly for fewer shares bought, and vice versa.

The market stage implements trading fees in order to raise the funds necessary to pay stakeholders when the event's outcome is realised, and to reward users for participation in the arbitration stage. Buying shares pushes the share price p towards \$1, while selling shares pushes it towards \$0, and the fee can be interpreted as a fee on the worst-case loss that an agent incurs. For a fixed parameter f, a "buy" transaction that pushes the share price to p incurs an additional charge of fp, while a "sell" transaction incurs a charge of $f(1 - p)$. Transactions are only charged a fee when the user increases their risk: if they are simply liquidating their position (selling shares they own or buying back shares they have sold) then no fee is charged. Users may trade in the market as long as they have enough funds to make the transaction and the deadline for the event has not passed. After the deadline has expired the users' positions are final and we move on to determining the outcome of the event in the arbitration stage.

4.1.2 Arbitration Stage

The arbitration stage is concerned with determining the perceived outcome of event X by taking reports from a subset of the users in the system, known as the arbiters. Each arbiter i receives a (private) signal $x_i \in \{0, 1\}$ that tells them the result of the event – this is analogous to reading the news, watching the match, or even hearing about it from a friend. They then submit a report $\hat{x}_i \in \{0, 1\}$ to the system that tells us what they believe the outcome to be. Note that since the signal they receive is private, we have no way to know whether the user is reporting what they truly believe, or whether they are trying to manipulate the system for their own gain. Arbiters are then paired randomly, and paired arbiters i and j are paid a reward of $u(\hat{x}_i, \hat{x}_j)$ according to the "1/prior with midpoint" mechanism, which we will detail below.

We can now determine the outcome of the market. In a traditional prediction market, users are paid \$1 for each share owned (or they must buy shares back at \$1 if they have shorted the security) if the event has a positive outcome, and \$0 otherwise. Hence if an event is almost certain to occur, more users will buy shares than sell them, as they expect they will be paid out for holding shares in the market when the outcome is realised. This will push the share price towards \$1. Similarly, if an event is believed to be unlikely, more users will sell shares than buy them and the price will approach \$0. This prevents arbitrarily large profits being made for risk-free bets. In our market the outcome of an event is the random variable $\hat{X} \in [0, 1]$ which is set to the proportion of arbiters that submit a report of $\hat{x}_i = 1$.

Stakeholders are then paid out in the usual way.

4.1.3 1/prior mechanism

We use the 1/prior payment mechanism to reward users for submitting reports on the outcomes of events, with a modification to incentivise truthful reporting. The 1/prior payment mechanism was conceived by Jurca and Faltings [12, 13] as a means of rewarding arbiters for participation in opinion polls, and Witkowski [19] generalises this mechanism to pay out different amounts depending on the signals reported by paired arbiters. For paired arbiters i and j with reports $\hat{x}_i$ and $\hat{x}_j$, the 1/prior mechanism pays i and j as a reward for reporting on the market outcome:

$$u(\hat{x}_i, \hat{x}_j) = \begin{cases} k\mu & \text{if } \hat{x}_i = \hat{x}_j = 0 \\ k(1-\mu) & \text{if } \hat{x}_i = \hat{x}_j = 1 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

in which k is a parameter and μ is the (common) prior belief that $X = 1$. A suitable value to use for μ would be the closing price for the market: if users feel an event is likely to happen the share price will be pushed towards \$1, while if they feel it is unlikely it will be pushed towards \$0.

The modification to the 1/prior mechanism is simple though requires two extra values to be computed. Let μ_1^i be the probability that, given that agent i receives a positive signal, another randomly chosen agent also receives a positive signal, and let μ_0^i be the probability that, given that agent i receives a negative signal, another randomly chosen agent receives a positive signal. We require the "update" probabilities μ_1, μ_0 to be common across all agents, which we can achieve by simply taking:

$$\begin{aligned} \mu_1 &:= \min_i \mu_1^i \\ \mu_0 &:= \max_i \mu_0^i \end{aligned} \tag{2}$$

The modified payment method we use, the "1/prior with midpoint" mechanism, is now simply Equation 1 with μ replaced by $(\mu_1 + \mu_0)/2$. This guarantees the incentives for arbiters are the same regardless of their signal.

4.2 Tools

The project is implemented in Common Lisp and the code has been developed and tested within the Steel Bank Common Lisp (SBCL) compiler and runtime system. Code version control has been achieved with Git and Github. Writing the web application has required the use of several libraries available from the library manager Quicklisp, specifically:

- Hunchentoot
- CL-WHO
- Mito
- SXQL
- Parenscript
- Smackjack

Hunchentoot provides the environment on which we host the server. Most importantly it provides automatic session handling, allowing for multiple users to be logged in at once, and easy access of GET and POST parameters, enabling interaction via HTML forms. To generate the webpages, we use CL-WHO, which converts Lisp statements into strings of valid HTML. Defining webpages while remaining in the Lisp environment means we may use Lisp's macro system to build abstractions for both defining structure and processing data in one interface.

Mito and SXQL provide the ability to connect to and interact with a Relational Database Management System (RDBMS). We use MySQL, though this choice is largely immaterial given our simple requirements of the database.

Parenscript incorporates Javascript into the site with the goal of improving user experience. Currently it is only used to ensure that all necessary fields during market creation, trading, and market resolution are non-empty to avoid sending incomplete data to the server. It will be used to a greater degree in the

future to ensure responsiveness: all information displayed to the user must be current to ensure that users are interacting with an up-to-date state of the system. We therefore plan to make more use of Parenscript and Smackjack, an AJAX library for Lisp, to allow for asynchronous interaction with the server. This will include, for example, continuously updating a stock's price or calculating the cost of a transaction without a page refresh, and stronger client-side validation.

References

[1] *Augur: the world's most accessible, no-limit betting platform.* `https://www.augur.net`. Accessed: 2020-02-16.

[2] *Hivemind*. `https://hvmd.io/`. Accessed: 2020-04-16.

[3] *IEM: Iowa Electronic Markets*. `https://iemweb.biz.uiowa.edu`. Accessed: 2020-02-16.

[4] *InTrade*. `https://intrade.com/`. Accessed: 2020-08-28.

[5] *Omen*. `https://omen.eth.link/`. Accessed: 2020-08-28.

[6] *Predictalot! (and we mean a lot).* `http://blog.oddhead.com/2010/03/05/predictalot/`. Accessed: 2020-02-16.

[7] *Predictit*. `https://www.predictit.org`. Accessed: 2020-02-16.

[8] *This new blockchain-based betting platform could cause napster-size legal headaches.* `https://www.technologyreview.com/2018/08/02/240354/this-new-ethereum-based-assassination-market-platform-could-cause-napster-size-legal/`. Accessed: 2020-08-27.

[9] M. Dudík, S. Lahaie, and D. M. Pennock, *A tractable combinatorial market maker using constraint generation*, in Proceedings of the 13th ACM Conference on Electronic Commerce, EC '12, New York, NY, USA, 2012, Association for Computing Machinery, p. 459–476.

[10] R. Freeman, S. Lahaie, and D. M. Pennock, *Crowdsourced outcome determination in prediction markets*, in Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI'17, AAAI Press, 2017, p. 523–529.

[11] R. Hanson, *Combinatorial Information Market Design*, Information Systems Frontiers, 5 (2003), pp. 107–119.

[12] R. Jurca and B. Faltings, *Incentives for expressing opinions in online polls*, in Proceedings of the 9th ACM conference on Electronic commerce - EC '08, Chicago, Il, USA, 2008, ACM Press, pp. 119–128.

[13] ——, *Incentives for Answering Hypothetical Questions*, in Workshop on Social Computing and User Generated Content - EC '11, 2011, p. 9.

[14] C. Kroer, M. Dudík, S. Lahaie, and S. Balakrishnan, *Arbitrage-free combinatorial market making via integer programming*, in Proceedings of the 2016 ACM Conference on Economics and Computation, 2016, pp. 161–178.

[15] Y. Liu, J. Wang, and Y. Chen, *Surrogate scoring rules*, in Proceedings of the 21st ACM Conference on Economics and Computation, EC '20, New York, NY, USA, 2020, Association for Computing Machinery, p. 853–871.

[16] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic Game Theory*, Cambridge University Press, USA, 2007.

[17] J. Peterson, J. Krug, M. Zoltu, A. K. Williams, and S. Alexander, *Augur: a decentralized oracle and prediction market platform*, arXiv preprint arXiv:1501.01042, (2015).

[18] J. Surowiecki, *The Wisdom of Crowds: Why the Many are Smarter Than the Few and how Collective Wisdom Shapes Business, Economies, Societies, and Nations*, Doubleday, USA, 2004.

[19] J. Witkowski, *Robust Peer Prediction Mechanisms*, PhD thesis, 2014.