4-2013

FORETELL: Aggregating Distributed, Heterogeneous Information from Diverse Sources Using Market-based Techniques

Janyl Jumadinova
University of Nebraska at Omaha

FORETELL: Aggregating Distributed, Heterogeneous Information from Diverse Sources Using Market-based Techniques

by

Janyl Jumadinova

A DISSERTATION

Presented to the Faculty of

The Graduate College at the University of Nebraska

In partial Fulfillment of Requirements

for the Degree of Doctor of Philosophy

Major: Information Technology

Under the Supervision of Dr. Prithviraj Dasgupta

Omaha, Nebraska

April, 2013

Supervisory Committee:

Qiuming Zhu

Ann Fruhling

Dora Matache

Kenneth Kriz

Abstract

FORETELL: Aggregating Distributed, Heterogeneous Information from Diverse Sources Using Market-based Techniques

Janyl Jumadinova, M.S., Ph.D

University of Nebraska, 2013

Advisor: Prithviraj Dasgupta

Predicting the outcome of uncertain events that will happen in the future is a frequently indulged task by humans while making critical decisions. The process underlying this prediction and decision making is called *information aggregation*, which deals with collating the opinions of different people, over time, about the future event's possible outcome. The information aggregation problem is non-trivial as the information related to future events is distributed spatially and temporally, the information gets changed dynamically as related events happen, and, finally, people's opinions about events' outcomes depends on the information they have access to and the mechanism they use to form opinions from that information. This thesis addresses the problem of distributed information aggregation by building computational models and algorithms for different aspects of information aggregation so that the most likely outcome of future events can be predicted with utmost accuracy. We have employed

a commonly-used market-based framework called a *prediction market* to formally analyze the process of information aggregation. The behavior of humans performing information aggregation within a prediction market is implemented using software agents which employ sophisticated algorithms to perform complex calculations on behalf of the humans, to aggregate information efficiently. We have considered five different yet crucial problems related to information aggregation, which include: (i) the effect of variations in the parameters of the information being aggregated, such as its reliability, availability, accessibility, etc., on the predicted outcome of the event, (ii) improving the prediction accuracy by having each human (software-agent) build a more accurate model of other humans' behavior in the prediction market, (iii) identifying how various market parameters effect its dynamics and accuracy, (iv) applying information aggregation to the domain of distributed sensor information fusion, and, (v) aggregating information on an event while considering dissimilar, but closely-related events in different prediction markets. We have verified all of our proposed techniques through analytical results and experiments while using commercially available data from real prediction markets within a simulated, multi-agent based prediction market. Our results show that our proposed techniques for information aggregation perform more efficiently or comparably with existing techniques for information aggregation using prediction markets.

Copyright

Dedication

I dedicate my dissertation to all of my family, near and far, who supported me each step of the way.

Acknowledgements

From the formative stages of this dissertation to the final draft, I owe an immense debt of gratitude to my advisor, Dr. Prithviraj(Raj) Dasgupta. His guidance, stimulating suggestions, willingness to listen, and immense knowledge helped me in all of my research and in writing of this dissertation. I could not have imagined having a better advisor and mentor for my Ph.D study.

Besides my advisor, I would like to thank the rest of my thesis committee: Dr. Zhu, Dr. Fruhling, Dr. Matache, and Dr. Kriz, for their encouragement, insightful comments, and hard questions. Special thanks are due to Dr. Matache for taking on my project. Our collaboration in 2011 paid off in Chapter 3.

I am grateful to have been a member of the C-MANTIC Research Lab directed by Dr. Dasgupta, and I am thankful for the funding I have received to support my doctoral research.

I also want to thank my fellow labmates in the CMANTIC group and my fellow PhD students for the stimulating discussions and fun activities.

Finally, I would like to give my greatest thanks to my parents Aliyakbar Jumadinov and Raibubu Yrsalieva, my sisters Nazgul and Asel Jumadinova and their families for supporting and encouraging me throughout my life, and for helping me to get through the toughest times; to my greatest loves my son Vincent and Oliman for always putting a smile on my face and helping me remember what is truly important. Without their unconditional support and love, this dissertation would never have been accomplished.

Contents

List of Figures

Chapter 1

Introduction

Forecasting the outcome of events that will happen in the future is an essential task in decision-making processes. Forecasting is especially difficult since the information related to future events frequently appears in bits and pieces, for example, as dispersed opinions, insights, and intuitions of people. Each person only knows a little but possible useful piece of information, and aggregating the dispersed information together may make considerable contribution to the decision making process. This is encountered in various domains such as forecasting the outcome of geo-political events, betting on the outcome of sports events, forecasting the prices of financial instruments such as stocks, and casual predictions of entertainment events. The uncertainty of the event changes as time goes by and new information related either directly or indirectly to the event becomes available. Also, with the number of different data sources and forms exploding, the age of "Big Data" has arrived; bringing with it challenges of analyzing the large amounts of data from various domains such as social networks, sensor networks, and healthcare, and making decisions based on it. Therefore, *how* to aggregate *vast* amounts of dispersed information *timely* for useful *decision making* is a crucial task.

1.1 Motivation

There are various forecasting methods that have been developed over the past decades. These methods can be roughly divided into statistical and non-statistical methods[16]. Statistical methods, such as econometric models and some machine learning techniques, require some historical data. The limitation of statistical methods is that in addition to requiring the existence of enough historical data, they also require that historical data contain valuable information about the future event. On the other hand, non-statistical methods frequently rely on experts' judgment and opinions. The limitation of non-statistical methods is that in order to elicit experts' opinions we have to identify experts, have them agree to participate, and determine how to combine different opinions when experts are in disagreement. All of these tasks are complex in their own way.

The reliability of collective decision making has been of substantial interest for a long time. In 1906 British scientist Francis Galton attended a weight judging competition at Plymouth. Almost eight hundred people estimated the weight of an ox. Some of the participants were experts at judging the weight of cattle (butchers, farmers), while others only used their intuition with no insider knowledge of cattle. Surprisingly enough for Galton, the mean estimate, which can be interpreted as the collective wisdom of the Plymouth crowd, was very accurate. Galton's experience suggests that under some circumstances, groups are remarkably intelligent and even smarter than the smartest people in them. In 2005 James Surowiecki coined the term the "wisdom of crowds" by describing how groups of people solve, under certain conditions, complex problems far better than single individuals[102]. There are

Figure 1.1: Prediction markets use the wisdom of the crowd and the market-based structure to predict the outcomes of unknown events.

various ways to utilize the wisdom of crowds such as using wikis, reputation systems, or polling mechanisms. Another way to aggregate dispersed information is by setting up a market. Economists have long understood that, in theory, the market prices in properly designed markets reflect the collection of all the information possessed by all the traders about future events[94]. Over the past few years, a market-based paradigm, called *prediction markets*, has emerged as an attractive market mechanism to solve forecasting problem using the aggregated opinions of the market's participants. A prediction market consists of human traders that bet their money on the possible future outcome of real-world events. The amount of money betted by a trader on a particular event depends on the trader's current belief about the outcome of the event. The aggregate value of the monetary bets made by different traders on an event dynamically determines the price of future bets related to the event. Prediction markets are considered to be an efficient aggregation mechanism for public opinion on the event because the dynamic price fluctuations of the bets related to an event is claimed to be an indicator of the public opinion or belief about the outcome of the event[109]. Its success is evidenced from the successful predictions of actual events done by prediction markets run by the Iowa Electronic Marketplace(IEM)[42], Tradesports, Hollywood Stock Exchange, the Gates-Hillman market[83], and by companies such as Hewlett Packard, Google[23] and Yahoo!. The effectiveness of prediction markets was further confirmed during the 2012 presidential election; they were much more effective than polls in predicting the vote shares of President Barack Obama and Gov. Mitt Romney. Prediction markets were initially introduced as social research tools for aggregating the opinions of a large number of people on the future outcome of imminent events. The following success of prediction markets as an effective aggregator of public opinion has led to their adoption in various domains ranging from academic research to commercial betting markets for popular events and predicting the performance or sales of products by software companies. Figure 1.1 shows some of the events that prediction markets have been used to predict. Although the concept of prediction markets originated from financial markets, ongoing research on this topic has evolved beyond purely market-oriented aspects into using prediction markets as an information aggregation and decision making mechanism.

Prediction markets have many advantages over statistical or non-statistical forecasting methods. Compared with statistical forecasting methods, prediction markets can incorporate real-time information, which was not contained in historical data. Compared with non-statistical forecasting methods, prediction markets are less constrained by space and time since there is no need to identify experts and solicit their participation, and hence are often less expensive in practice, and they do not need to deal with conflicting opinions. More importantly, prediction markets can potentially make real-time predictions that take advantage of the dispersed information, which are sometimes hard to capture using other forecasting methods.

1.2 Thesis Objectives

Despite the advantages and recent popularity of prediction markets, understanding and explaining the behavior of prediction markets and how to design effective prediction markets are still open questions to a large extent. If prediction markets are to be used to assist in making critical decisions in the real world, investigating these questions is imperative. The goal of this thesis is to provide an understanding of the properties and performance of prediction markets, through rigorous theoretical and empirical examinations, and to obtain an initial framework to guide prediction market design and development for decision making. To do this, we divide the thesis into five distinct contributions.

Next, we discuss the main contributions of this thesis.

1. Analyzing the Effect of Information Related Parameters on Prediction Markets. The first contribution is a multi-agent system that is used to analyze the effect of information on the prediction market performance. The effect of information on prediction markets is a crucial factor that affects the behavior of the trading agents in the market. Information about an event that the trading agents receive also influences the prices corresponding to the event and finally determines the outcome of the event. Our multi-agent based system incorporates different information-related aspects including the arrival rate of the information, the reliability of the information, the penetration or accessibility of the information among the different trades and the perception or impact of the information by the trading agents. The multi-agent implementation of a prediction market allows us to easily analyze and verify the trading agents' behavior while varying different market and agent related internal parameters of the prediction market, as well as external parameters related to the information about events arriving at the market.

2. Analyzing the Behavior of the Trading Agents. Researchers have proposed theoretical models capturing individual aspects of prediction markets such as utility theory-based models for participants' behavior, or aggregation strategies for combining the information from the market's participants[13;18;81]. However, behavior of the prediction market's participants on the market's predicted outcome in a partially unknown environment has not yet been fully investigated. For the second contribution we address this deficit by developing a game theoretic representation of the trading agents' interaction and determining their strategic behavior using the equilibrium outcome of the game. We also consider risk preferences of the agents and show theoretical properties for truthful revelation from risk averse agents. We empirically compare this equilibrium trading strategy with five different trading strategies used in similar markets with a detailed commercially available data from the Intrade prediction market[51] that contains time-stamped information on individual trades of each trader buying and selling securities as well as on traders' volume and their duration in the prediction market.

3. Analyzing the Dynamics of the Prediction Market. Despite a growing research on prediction markets, their implementation in practice is still difficult. It is important to know under what conditions the prediction market becomes most efficient. As the third contribution we focus on the dynamics of prediction markets under various conditions using Boolean Network techniques. Using a Boolean Network and a mean-field approaches from statistical physics we generate a mathematical model for a prediction market. In our model one node represents the market maker, that at each time step aggregates the information from the other nodes in the system, that represent the trading agents. Then, using the tools from dynamical systems and chaos theory[?], we analyze the evolution of the aggregated information under various scenarios.

4. A Prediction Market used for Decision Making in Multi-Sensor Domain. For the fourth contribution we study a decision making setting that uses a prediction market framework. Accurate information aggregation about uncertain events is very important for a decision maker. Within the last decade, much empirical and analytical work on prediction markets has shown prediction markets to be a successful information aggregation mechanism. We consider an analogous problem of information fusion from multiple sensors of different types with the objective of improving the confidence of inference tasks, such as object classification. We develop a multi-agent prediction market-based technique to solve this information fusion problem. We experimentally verify our technique using simulations of multi-sensor information fusion for an automated landmine detection scenario.

5. Distributed Prediction Markets. For the fifth contribution we consider a novel, yet practical setting of prediction markets called distributed prediction markets, where the predicted outcome of an event in one prediction market is affected dynamically by the predicted outcome of similar events in other, simultaneously running prediction markets. We focus on the problem of decision making facing a market maker to determine the possible outcome of an event within such a setting. We propose a formal framework based on graphical games to model the distributed prediction market setting and to capture the local interactions between multiple market makers. We also develop an algorithm that determines the best action for the participants in the prediction market using our proposed model. Our results show that our agent-based distributed prediction market technique operates more efficiently and accurately than individually-operating, isolated prediction markets in predicting the event's outcome. Table 1.1 shows the specific research questions that we address in each of our contributions. We investigate these questions from four approaches: theoretical examination, empirical analysis, and design and development. Theoretical examination can help us understand how and why prediction markets behave a certain way. This is achieved through developing and analyzing computational models of prediction markets. Empirical analysis using some real-world data and theoretical models aims at analyzing the behavior of prediction markets. Based on previous results, we then investigate issues of design and development of information aggregation system for decision making using prediction

Contr.	Research Question	Research Approach
1	How do changes in different aspects of information affect the behavior of prediction markets?	Empirical Analysis
2	How do different trading agent behaviors affect the behavior of prediction markets?	Empirical Analysis
2	What trading strategies perform the best in prediction markets?	Empirical Analysis
2	How can prediction markets incentivize trading agents to participate in order to achieve a higher prediction accuracy?	Theoretical Examination
2	How does the trader's behavior using a formal game-theoretic model compare to the trader's behavior in real prediction markets?	Empirical Analysis
3	How does a prediction market evolve and what are its dynamics under different market and trader conditions?	Theoretical Examination Empirical Analysis
3	How can we make a prediction market unaffected by "noise"?	Theoretical Examination Empirical Analysis
4	How can prediction markets be used for decision making?	Design and Development Theoretical Examination
4	Is there an advantage of using prediction markets for sensor fusion?	Empirical Analysis
4	Can a prediction market-based model for sensor fusion be effective in realistic settings given various limitations?	Empirical Analysis
5	How can we design a distributed model of a prediction market to be used for decision making?	Theoretical Examination Design and Development
5	Is our distributed model of the prediction market incentive compatible?	Theoretical Examination
5	How does a distributed information aggregation in a prediction market compare to the centralized aggregation?	Theoretical Examination Empirical Analysis

Table 1.1: Research questions that are addressed in this thesis.

market.

1.3 Thesis Organization

This thesis is structured in six chapters. Chapter 2 introduces basics of prediction markets and reviews related work extensively. The rest of the chapters discuss contributions 1 through 5 and answer corresponding research questions. Chapter 3 analyzes the effect of different information related parameters on the traders' behavior and on prediction market's performance. Chapter 4 studies the trading agents' interaction and proposes an algorithm for their strategic behavior. Chapter 5 studies the dynamics of prediction markets under various conditions using Boolean Network techniques. Chapter 6 solves an information fusion problem from multiple sensors of different types using prediction market-based framework. Chapter 7 proposes distributed prediction markets framework where multiple prediction markets running similar events can affect each other's aggregated prices. And finally, chapter 8 concludes the thesis, discusses some future work, and presents some open problems.

Chapter 2

Related Work

In this chapter, we highlight the major participating entities and essential features of the operation of a prediction market. We then summarize the related work on prediction market.

2.1 Prediction Markets Overview

2.1.1 Prediction Market Participants and Operations

Prediction markets[109] use the collective opinions of a group of people, called the market's *traders*, to forecast the outcome of a future event. Usually, a payment mechanism with either real or play money is used to motivate the traders to participate in the market. The prediction market has a set of future *events* whose *outcome* has not yet been determined. Traders bet their money on the possible future outcome of real-world events, such as the presidential elections, some movie's box office performance, or the sales performance of a particular product. The amount of money betted by a trader on a particular event depends

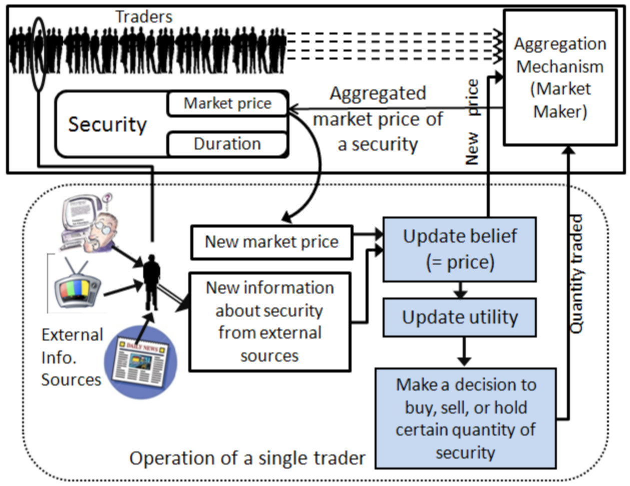


Figure 2.1: Essential components of a prediction market and operations of a trading agent in a prediction market.

on the trader's current belief about the outcome of the event. The outcome of each event is considered as a binary variable with the outcome being 1 if the event happens and the outcome being 0 if it does not. Each outcome of an event has a security associated with it. A *security* is a specific contract that has a *market price* associated with it and yields payments based on the outcome of an uncertain future event. One example of a security could be "Democratic Party candidate to win the 2016 US Presidential Election". Securities can be purchased or sold by traders at any time during the lifetime of the security's event. A security expires when the event associated with it happens at the end of the event's duration. At this point the outcome of the event has just been determined and all traders are notified of the event's outcome. The traders that had purchased the security during the lifetime of the event then get paid a certain amount of money for every security they possess if the event happens with an outcome of 1, or, they do not get paid anything and lose the money they had spent on buying the security if the event happens with an outcome of 0. During the lifetime of the event, the aggregate value of the monetary bets made by different traders on an event dynamically determines the market price of a security associated with the outcome of the event. The market price of a security in a prediction can be interpreted as the probability that the event's outcome associated with that security will be realized. The institution or company running the prediction market usually performs the aggregation process and determines the market price using an intermediary called the *market maker*. Traders interact with the market maker to buy and sell securities. The market maker aggregates the prices at which traders want to trade their securities into a single market price that represents the possible outcome of the event. Figure 2.1 shows the main entities of a prediction market along with the operations performed by a trader in a prediction market.

A special class of prediction markets is a combinatorial prediction market[46]. In a combinatorial prediction market, allowable bets are not explicitly listed, but are rather implicitly defined on a combinatorial space. For example, in a market for NBA basketball rankings involving n teams, the outcome space might be all the $n!$ possible permutations of the teams. And the allowable bets could be permutations such as "Los Angeles Lakers wins the NBA championships" or "Los Angeles Lakers beats Dallas Mavericks and Chicago Bulls". As another example, in a market for the state-by-state outcome (win or loss) of the US presidential elections, the outcome space could be 2^{50} possible state-by-state results, while allowable bets are Boolean statements like "Obama wins in Ohio and California but not in Nebraska". Combinatorial prediction markets offer greater expressiveness by being able to capture complex relationships between events. However, having to work with a combinatorial outcome space also increases the complexity and computational cost of predicting outcomes in a combinatorial prediction market.

2.1.2 Prediction Markets and Early Models

Prediction markets were started in 1988 at the Iowa Electronic Marketplace to investigate whether betting on the outcome of geo-political events (e.g. outcome of presidential elections, possible outcome of international political or military crises, etc.) using real money could elicit more accurate information about the event's outcome than regular polls. Following the success of this experiment, several other prediction markets have been started in different domains. For example, the Hollywood Stock Exchange[50] is used to predict performance of movies at the box-office, Betfair[6] is used to predict the outcome of sports or finance-related events, while Intrade[51] is used to predict outcomes of various events ranging from political to entertainment. Recently, several companies such as Google, Hewlett Packard, Microsoft and Yahoo! have used prediction markets to analyze different technology trends such as information processing practices in organizations and product management strategies. In 2003, the interest in prediction markets was strong enough that a think tank within the U.S. Department of Defense planned to establish a Policy Analysis Market where securities corresponding to different political and geographical events could be traded.

With the successful operation of several prediction markets, researchers attempted to make mathematical models to understand various aspects of the market's operation. Among the earliest research on this topic, Wolfers and Zitzewitz[109] compared forecasts made by prediction markets in different domains such as oil markets and movie box office earnings to show that prediction market forecasts follow other independent forecasting mechanisms very closely. The major insights offered by their work hypothesized that prediction markets will not perform well at predicting small probability events, that the profit motive of traders is sufficient to prevent prediction markets from being manipulated, and, that it was important to design the market trading rules carefully. Contrary to this result, Manski[69] concluded that the predictions of event outcomes made in prediction markets did not closely correspond to the actual probability beliefs of the traders in the market unless the probability is near either 0 or 1. In response, Wolfers and Zitzewitz[111] presented empirical results showing that prediction market prices closely track the mean beliefs of traders, providing a foundation for the claim that predictions markets efficiently aggregate beliefs. The authors used different mathematical belief functions to show that distributions of beliefs that are not too diverse result in market prices that are close to the mean belief of the traders. With the support of empirical prediction market data from U.S. election and professional football domains, the authors concluded that Manski's special case that showed how prices and beliefs may diverge was not the best case scenario but, rather, a worst-case scenario. Further, Gjerstad[34] showed that traders' risk aversion and beliefs may significantly affect the equilibrium price. Using the same coefficient of relative risk aversion (CRRA) expected utility function but independent wealth and belief distribution for the traders, Gjerstad showed that predictions are typically very close to the mean belief of the traders in the market if the distribution of beliefs is smooth and risk aversion is modeled into the traders' behavior.

Despite encouraging results in these earlier papers motivating prediction markets, other evidence, however, suggests that the relative performance advantage of prediction markets may be small, and that markets may not even be the best performers. In predicting the outcome of football games, Chen *et al.* reported that pooled expert assessments are comparable in accuracy to prediction markets[15]. Goel *et al.*[36] compared the performance of prediction markets against conventional methods, i.e. polls and statistical models for sports and entertainment domains and showed that prediction markets do not significantly outperform the other methods. Therefore, researchers have been tackling important open questions that help to identify conditions under which prediction markets work the best, how they work, and how to design efficient prediction markets. We review some of these techniques in the next section.

2.2 Prediction Market Models

In recent times, with the advent of intelligent software agents that can perform complex computations to model the behavior of human traders in a prediction market, much of the prediction market research has concentrated on using artificial prediction markets where software agents mimic the functionality of human traders and market makers from a real prediction market. Clearly, the advantage of artificial, agent-based prediction markets is that mathematical models of trader behavior and market operation can be studied and analyzed relatively easily by simulating the operations in the market while using data from real prediction markets. Because of the widespread use of agent-based prediction markets, in the remainder of this chapter we have used the terms trader and *agent* interchangeably to refer to either a human trader or a software trading agent.

2.2.1 Prediction Market Trading Protocols

Because prediction markets have their roots in financial markets, a popular protocol for trading in financial markets called the continuous double auction (CDA) protocol, has been used relatively extensively in early work on prediction markets[92;109]. In a CDA each security is associated with two prices in the market - a *bid* price and an *ask* price. The bid price is the price at which agents are willing to purchase the security and the ask price is the price at which agents are willing to sell the security. The difference between bid price and sell price is known as the *spread*. There are usually two types of orders that agents can place, known as *market orders* and *limit orders*. Market order is an order to buy or sell a security at the best available price and is of the form "Buy(Sell) X shares". Market orders are executed immediately at the prevailing market bid or ask price. Limit order is an order to buy or sell a security at a specific price and is of the form "Buy(Sell) X shares at price Y". The highest buy limit order and the lowest sell limit order form the market bid and ask prices respectively. A new market buy order is traded at the price corresponding to the lowest limit sell order, while, a market sell order is traded at the price corresponding to the highest limit buy order. Traditional Iowa Electronic Markets and InTrade prediction markets use the CDA protocol. The CDA protocol offers two main advantages over other common prediction market trading protocols. First, with CDAs, the price of a security is allowed to make instantaneous jumps by arbitrary values. These price jumps can immediately capture the extent of the impact of an external event or information on a security's price. For example, following a natural calamity affecting crude oil production, a security corresponding to the event 'crude oil futures closing above $100 in 2013' can go, for example, from $90 to $40 in one trade in a prediction market using CDA. Secondly, agents using CDA are able to express their opinions precisely by placing bid or ask orders at specific prices. In contrast, other protocols require agents to specify the quantity they wish to trade while regulating the market price, usually through the market maker. However, the CDA protocol has shortcomings when used in a prediction market. In the CDA, traders reveal the prices at which they wish to trade through buy and sell orders, and, consequently, run the risk of potentially losing profit by revealing their willingness to trade beforehand to other traders. Another major drawback of the CDA protocol is that it fails to work if there is insufficient liquidity, that is, if there are very few traders who cannot reach an agreement on a trading price, and, thus may never trade securities with each other. Liquidity problems are most prominent in combinatorial prediction markets since they have vast numbers of outcomes to predict. Hanson[46] proposed the use of automated market makers within the context of combinatorial prediction markets to deal with matching and liquidity problems using a market scoring rule, which we discuss next.

2.2.2 Automated Market Makers and Scoring Rules

Scoring rules were first proposed by Brier[11] and analyzed further by Winkler[108]. They were initially used for forecasting weather and finance events. A scoring rule is a mathematical formula that determines the payment received by an agent for its forecast for the outcome of an event. Suppose an event e has m possible outcomes with o_i representing the i-th outcome. $\mathbf{r} =< r_{o_1}, r_{o_2}, ..., r_{o_m} >$ is a reported probability estimate vector where r_{o_i} corresponds to the price that a risk-neutral trader wants to pay for securities corresponding to outcome o_i. A *scoring rule* $\mathbf{S} = \{s_1(\mathbf{r}), s_2(\mathbf{r}), ..., s_m(\mathbf{r})\}$ assigns a score $s_i(\mathbf{r})$ to a trader who reports $\mathbf{r}$ if outcome o_i is realized *. Scoring rules have been shown to elicit good probability estimates from individual traders.

A regular scoring rule (if $s_i(\mathbf{r})$ is finite when $r_i > 0$) is **(strictly) proper** if truthful reporting (strictly) maximizes the expected score of a risk-neutral agent, that is, given the true belief $\mathbf{p}$, where $\sum_i p_i = 1$,

$$E_p(s_i(\mathbf{p})) \geq E_P(s_i(\mathbf{r})) \text{ given} \sum_i r_i = 1 \tag{2.1}$$

s_i is strictly proper if the inequality in Equation 6.8 is strict.

Hanson[46] combined features of scoring rules and prediction markets into a market scoring rule (MSR). A special type of scoring rule called the *logarithmic scoring rule* (LSR), which has been used extensively in prediction markets, deserves special mention. A

*The specific mathematical formula for $s_{o_i}(\mathbf{r})$ is determined by the type of the scoring rule used, such as, quadratic, spherical, logarithmic, etc.

logarithmic scoring rule is defined by the following expression:

$$s_i(\mathbf{r}) = a_i + b \cdot log(r_i), \tag{2.2}$$

where, $s_i(\cdot)$ is the score corresponding to the i-th outcome of an event being realized given a report $\mathbf{r}$. $a_i > 0$ is an arbitrary parameter while $b > 0$ is a parameter determined by the market maker to control its monetary risk as well as the maximum quantity of securities that agents can buy or sell at or near the current price without causing massive price swings. Larger values of b allow agents to trade more frequently but also increase the market maker's chances to lose money. The logarithmic scoring rule has some desirable properties - it preserves the conditional independence relations* [47], and it is the only rule that can be used both to reward an agent and to evaluate its performance using statistical likelihood methods[108].

When the general logarithmic scoring rule is used sequentially by agents to calculate their payoffs in a prediction market, the scoring rule is called the Logarithmic Market Scoring Rule (LMSR). The LMSR gives a formula to calculate the current market price for a small quantity (small enough so that the trade can be completed in one time period) of a security corresponding to the event's outcome being o_i, as follows:

$$p_i(\mathbf{s}) = \frac{exp(\frac{s_i - a_i}{b})}{\sum_{j=1}^{m} exp(\frac{(s_j - a_j)}{b})},$$

where, $\mathbf{s}$ is the logarithmic market scoring rule vector over all possible outcomes of the event, $s_i \in \mathbf{S}$ and other parameters are as defined in Equation 2.2.

A market maker using the LMSR to determine the market price in a prediction market is called an LMSR market maker. As pointed out by Hanson[46;47], the fundamental difference in the trading protocol using an LMSR market maker from that using a CDA is that agents now only express quantities they wish to buy or sell at the current market price that is set by the market maker, instead of specifying the price along with the quantity at which they

*Placing a bet on conditional event A given event B should not change the conditional probability distribution of B or C for some event C that is unrelated to how event A might depend on event B

wish to buy or sell through buy and sell orders. An LMSR market maker is a type of market maker called a cost-function based market maker[13]. It offers m securities, one for each possible outcome, where each security pays \$1 if the corresponding outcome happens and \$0 otherwise. Let $\mathbf{q} = (q_1, q_2, ..., q_m)$ denote the vector specifying the number of units of each security held by different agents. The market maker first calculates a cost function to reflect the total money wagered in the prediction market by the agents. Agents inform the quantity of a security they wish to buy or sell to the market maker. If an agent purchases δ units of a security, the market maker determines the payment the agent has to make as $C(\mathbf{q}+\delta) - C(\mathbf{q})$. Correspondingly, if the agent sells δ quantity of the security, it receives a payoff of $C(\mathbf{q}) - C(\mathbf{q}-\delta)$ from the market maker. For the LMSR market maker, the cost and price (corresponding to outcome o_i) functions are given by:

$$C(\mathbf{q}) = b \cdot log(\sum_{i=1}^{m} e^{q_i/b}), \text{ and,}$$

$$p_i(\mathbf{q}) = \frac{exp(q_i/b)}{\sum_{j=1}^{m} exp(q_j/b)}, \text{respectively,}$$

where $\sum_{i=1}^{m} p_i(\mathbf{q}) = 1$.

The LMSR market maker has been extensively used in recent prediction market research[1;13;14;43;78;83;87] where it has been implemented, extended, analyzed or compared with other market maker strategies. It is also used at several prediction markets, such as Inkling Markets, Consensus Point, Yahoo!, Microsoft, BizPredict, and Washington Stock Exchange. Among recent significant works on LMSR market makers, Chen and Vaughan[14] showed that there is a one-to-one mapping between the strictly proper market scoring rules and convex cost-function based market makers. In particular, they show that these two formulations map to each other if and only if

$$C(\mathbf{q}) = sup_{\mathbf{p}\in\Delta_m} \left(\sum_{i=1}^{m} p_i q_i - \sum_{i=1}^{m} p_i s_i(\mathbf{p}) \right),$$

where q_i is the number of units of security corresponding to outcome o_i that is being held by all agents, p_i is the true belief of an agent that the outcome is o_i, and Δ_m is the probability simplex that consists of all possible probability distributions for a random variable with state space $[m] = \{1, 2, ..., m\}$. The authors' mapping allows for an easy conversion between market scoring rule and cost-function based market makers. They also showed that a cost-function based market maker with bounded loss can be interpreted as a no-regret learner, that is as learning a probability distribution over outcomes by treating each observed trade as a training instance.

The computational complexity of an LMSR market maker has been a topic of interest in several literature. In [17], the authors examined permutation combinatorics, where outcomes are permutations of objects, and, Boolean combinatorics, where outcomes are combinations of binary events, within a combinatorial prediction market. The problem of LMSR pricing in the combinatorial prediction market was shown to be $\#P$ hard, even with severely limited languages. They also proposed an approximation technique for pricing in combinatorial prediction markets that is based on permutation learning technique.

One way to deal with this computational complexity is to use a restricted betting language that limits the set of tradable securities. Guo and Pennock [43] studied combinatorial prediction markets with an LMSR market maker where events are represented as a tree-like hierarchy and agents can bet on the sum of values at any tree node. For example, in the US presidential election market, the total number of electoral votes determines the outcome of the election. The tree for this example has "Overall Election Results" as a root node and each state's result as a leaf node. Therefore, betting on the number of electoral votes won by President Obama from a set of states is the same as betting on the weighted sum of the binary variables representing those states, where weights are electoral votes of different states. The authors proposed three expressive languages for betting on the weighted sum:

- Sum of arbitrary subset (SAS): betting on the weighted sum of an arbitrary subset of values
- Sum of varying weights (SVW): agents set their own weights in their bets but restrict subsets to form hierarchy
- Sum with predefined weights (SPW): betting on the weighted sum of selected subsets

of values, where weights are predefined and subsets form a hierarchy

They further analyzed the computational complexity of each proposed language and showed that pricing for SAS and SVW was NP-hard, but a polynomial time algorithm existed for SPW.

Researchers have also recently identified some drawbacks of the LMSR market maker and proposed alternate market making rules. Othman and Sandholm[83] ran a prediction market with an automated market maker using LMSR to predict the opening day of the Gates and Hillman Centers at Carnegie Mellon University. They found two flaws in LMSR automated market maker - spikiness of prices across similar events, and, liquidity insensitivity leading to price volatility. To fix these two problems, Othman *et al.*[87] introduced a modified LMSR market maker. The authors argued that parameter b was non-trivial to set since it controlled how quickly prices move and also the worst-case loss of the market makers. Their proposed market maker increases b continuously with the total number of all securities purchased by agents thereby. Thus, the modified market maker automatically adjusts how easily market prices change according to the amount of trading in the market. The authors showed that no market maker can satisfy three properties: (1) *path independence(budget-balanced)*, (2) *no-arbitrage* (no gain without a risk), (3) and *liquidity sensitivity* (small trades move market prices less in thick(liquid) markets). They noted that an LMSR market maker satisfies the path independence and no-arbitrage properties. They have subsequently weakened the no-arbitrage property in the LMSR market maker and designed an automated market maker that can run at profit.

Brahma *et al.*[10] evaluated two different market makers - an inventory-based logarithmic market scoring rule (LMSR) market maker and a Bayesian market maker (BMM). The authors used an information-based market maker model that is based on the canonical Glosten-Milgrom model[35] of price-setting under asymmetric information. The market maker has to set a bid (buy) price, denoted by bid, and an ask (sell) price, denoted by ask. It does this using its beliefs about the realization of outcome o_i, such that its expected profit is non-negative. An agent receives a signal s, where the variance of s measures the uncertainty in the agent's signal, and trades as follows: if $s < bid$, the agent sells, and if

$s > ask$, the agent buys. The authors found that BMM offered market price stability and lower expected loss compared to LMSR market maker, but BMM did not guarantee bounded loss. The authors also identified that LSMR market maker represents an uninformed agent and may be outsmarted by human agents; BMM market maker on the other hand, is able to handle large trades and adapt quickly to market shocks.

Dudik *et al.*[29] proposed a new automated market maker for providing liquidity across multiple logically inter-related securities and propagating as much information as possible among logically related securities while keeping the pricing algorithm tractable. Their model consists of two components. First, a large prediction market is broken into small markets, which solves the intractability problem, but may introduce arbitrage opportunities. Secondly, a general scheme is introduced that detects and removes arbitrage based on convex optimization and constraint generation using linear equalities and inequalities. The authors found that their method lies between a market maker that treats related securities as independent and unrelated, and, a full combinatorial market maker for which pricing is computationally intractable. They also found that their market maker is competitive with an LMSR market maker but it adds the ability to scale to exponential outcome spaces.

So far we have discussed automated market makers that satisfy some of the four properties that are desired to have in a market maker - bounded loss, the ability to make a profit, a vanishing spread that approaches zero in the limit as volume gets large, and, unlimited market depth, that is, the price of any fixed-size trade approaches the marginal bid or ask price. Satisfying all four characteristics is difficult because several of the properties are contradictory. For example, different variations of the LMSR market maker that we discussed above[13;14;83] can offer bounded worst-case loss and no marginal bid/ask spread. But they do not offer the ability for the market maker to make a profit, and they only offer a fixed liquidity of the market, i.e. the market does not become more liquid with the increase of the volume. Othman and Sandholm[86] presented a market maker that satisfies all four of the desired properties. Their proposed market maker extends the constant-utility cost function market maker[13] with two separate functions, *liquidity* and *profit*, that are added to the market prices that are quoted to the agent. The *liquidity function* uses its gains to increase the amount of liquidity provided by the market maker, while the *profit function* represents a "lockbox" of savings that is separate from the rest of the market maker's decision-making process. For most of the market makers in the literature, including the LMSR market maker, an agent can buy and then immediately sell a security from the market maker at no cost. The authors showed that real markets do not generally function this way because an agent that buys and then immediately sells a security from the market maker will encounter a loss. Their proposed market maker imitates real world market makers in their path dependence. The authors provide the complete set of conditions that are required for different parts of their automated market maker along with several examples of functions that satisfy this set of conditions, noting that any mix of these examples will produce an automated market maker with certain desirable qualities.

2.2.3 Dynamic Parimutuel Markets

Complementary to using MSRs, Pennock[91] proposed another model for prediction markets, called a dynamic parimutuel market (DPM). Under this model, which is similar to horse race betting, market liquidity is automatically created by allowing all agents to purchase any security at any time. There are m securities in a DPM, each corresponding to one outcome. The agents that participated in the prediction market split the total pool of money at the end of the prediction market in proportion to the amount they have put in. The agents are allowed to see the securities they hold before the outcome is known since the market price of a single security varies dynamically according to a price function. DPM acts as an automated market maker similar to the cost-function based market maker, except that the payoff of a security is not fixed.

The common cost and price functions used in DPM are:

$$C(\mathbf{q}) = \sqrt{\sum_{i=1}^{m} q_i^2} \text{ and } p_i(\mathbf{q}) = \frac{q_i}{\sqrt{\sum_{j=1}^{m} q_j^2}} \text{ respectively.}$$

The payoff of a security, corresponding to outcome o_i when outcome o_i happens and $\mathbf{q}$ is the quantity vector at the end of prediction market, is given by:

$$r_i(\mathbf{q}) = \frac{C(\mathbf{q})}{q_i}$$

Unlike a regular parimutuel market, in Pennock's DPM, agents are allowed to trade their previous investments using a continuous double auction (CDA) protocol. However, such trading occurs at a heavily discounted market price. This creates considerable (monetary) risk for agents to invest in bad securities. Thus a DPM makes rational agents inherently risk-averse or risk-neutral, which makes prediction markets operate more efficiently.

Recently, there has been some interest in comparing and unifying market scoring rules and DPM mechanisms for prediction markets. Nikolova and Sami[78] developed a new abstract betting game, called the *segment game*. A segment game can serve as a strategic model of DPMs, and can also model the strategies in market scoring rules. The segment game is tractable to analyze, and has an attractive geometric visualization that makes the strategic moves and interactions more transparent. The authors used their segment game to prove several strategic properties about the dynamic parimutuel market. They also proved that a special form of the segment game is strategically equivalent to the spherical scoring rule, and, strategically similar to other scoring rules.

2.3 Models for Overall Prediction Market Behavior

Trading protocols in prediction markets which were reviewed in the previous section specify rules for determining trading prices by the traders or by the market maker. In parallel, some recent work on prediction markets analyzes the market's overall price dynamics while using the trading protocols mentioned earlier. One of the earliest computational models of a prediction market[32] used a Shapley-Shubik game to formally represent the market's interactions. In the Shapley-Shubik model trading occurs in rounds and each agent must offer at least one security for sale with no restrictions on credit. The market price of a security at the end of each round is determined by averaging the bids over the number of agents. Agents bid for securities on each round and agents are always assumed to reveal their true preferences. The authors found that good predictions depend on some knowledge of the market by the agents but knowledge alone is not sufficient. Instead, they argued that the prediction market structure that affects how private information is shared is important and the design of prediction markets that converge to direct communication equilibrium are an important research topic. Chen *et al.*[16] generalized the Shapley-Shubik game based model to allow for aggregate uncertainty, which occurs when the state of the world is not known even after all agents have shared their information. Their work addresses different properties related to the market equilibrium in prediction markets such as convergence to a consensus equilibrium, the rate of convergence, identifying the best possible equilibrium, and, whether a prediction market is guaranteed to converge to the best possible equilibrium.

Section 2.4 also describes various models that were developed to study the behavior of the agents and their effect on the prediction market.

2.4 Traders: Behavior and Strategies

There have been a number of prediction market studies that analyzed real prediction market data to identify some properties of the behavior of the agents in prediction markets. For example, Cowgill *et al.*[23] analyzed the trade data from internal prediction markets at Google. While other companies have created such markets, the Google markets were the largest such experiment at the time. The study had the advantage of having trade-by-trade data and short-lived markets. 157 different markets were analyzed and all had the common characteristic of representing outcomes that had a direct impact on Google. Measures of geographical, organizational, and social proximity and demographic similarity were used to study trading behavior and to investigate correlations in information and opinion. These measures were used in a regression model designed to predict the differences in the holdings of individual agents after each trade. The prior positions of colleagues were used to predict the size and direction of each trade. The authors found that the agents tended to be most optimistic about subjects over which they had some control, such as completing a project on time or whether a particular office would be opened. Their results also suggest that micro-geography played an important role in the flow of market information - trades of physically proximate employees were correlated after the employees began to sit next to each other. And finally, work history and organizational proximity played lesser roles while social connections and demographics had little explanatory power.

Another aspect investigated by researchers is whether agents in a prediction market respond to play money in the same manner they do to real money. To investigate this, Servan-Schreiber *et al.*[98] conducted an online experiment comparing the predictions of TradeSports.com (real money) against predictions of NewsFutures.com (play money) regarding American Football outcomes during the 2003 − 2004 NFL season. The authors suggested that real-money markets may better motivate information discovery while play-money markets may yield more efficient information aggregation.

There have also been a lot of theoretical studies on the behavior of the agents in prediction markets. For example, Othman and Sandholm[85] studied the behavior of simple agents and how their behavior affect market prices over time. They considered two models: adversarial ordering model and random ordering model. They used a pricing rule that defines a structured way of adjusting prices in response to instantaneous aggregate demand. They assumed that the market maker's pricing rule is a strictly increasing function, which is history independent. They also assumed that the simple agents have fixed beliefs, fixed budget, participate only once and sequentially interact with an automated market maker. They found that under the adversarial ordering model, market prices can be arbitrarily uninformative and show that final market prices diverge arbitrarily based on the order of agent participation, while under random ordering model market prices are informative. They proved their results for two agents and then for a countable set of agents. Their results suggest that a market can become unhinged - as a failure of the 1996 Iowa Electronic Market vote-share market, and agent ordering has the greatest potential of skewing the market prices at the end of trading. This work, however, assumed that agents are myopic and risk-neutral. These assumptions have been relaxed in other literature, which are discussed next. account.

2.4.1 Incentive Compatibility and Bluffing

Hanson[46] has shown that, for risk-neutral agents that are myopic (i.e., do not account for the effect of their trades on other agents), it is optimal for each agent to reveal its true beliefs of the outcome of the traded event in a market scoring rule-based prediction market. This result leads to two questions: 1) What happens when agents take into account future payoffs, i.e. when they are non-myopic? 2) What happens when agents are not risk-neutral?

The first question is addressed in Chen *et al.*[18], where the authors studied whether there exists game-theoretic equilibria at which agents reveal their truthful information quickly (as soon as they can). They considered one event in a prediction market with two outcomes using an LMSR market maker and risk-neutral agents. They modeled such a prediction market as an n-player, incomplete-information, dynamic game. At the beginning of their prediction market each agent i gets a private signal s_i. The joint distribution of s_i's and the outcome of the event is common knowledge. Agents trade in the prediction market according to a predefined sequence. The authors analyzed two scenarios: when agents have conditionally independent signals and when agents have unconditionally independent signals. Agents' signals are *conditionally independent* when their observations are independent given the true state of the world. For example, signals are conditionally independent in the prediction market trying to predict whether a product was manufactured with high quality materials or low quality materials. Signals are *uncondionally independent* when signals may influence the outcome of the event, but are not caused by the true state of the world. For example, signals are unconditionally independent in a presidential election prediction market. The authors showed that the truthful strategy is an equilibrium strategy when agents have conditionally independent signals, and, a mixed strategy of *bluffing* with a certain probability is an equilibrium strategy, when agents have unconditionally independent signals, where bluffing is the strategy of betting contrary to one's information in order to deceive future agents, with the intent of gaining profits on their resultant misinformed trading.

Jian and Sami[53] extended the work in[18] by conducting experiments with humans to study the speed and efficiency of information aggregation. They compared two market scoring rules - direct[46;47], where agents report their beliefs as probabilities, and, indirect[13], where agents reveal their beliefs through buying and selling. They found that the information is better aggregated in the experiments when the trading sequence is prespecified. They also found that under the assumption of a pre-specified trading sequence, agent behaviors are different under the two information structures - conditionally independent signals and unconditionally independent signals. When the trading sequence is prespecified there are more manipulative behaviors with unconditionally independent signals than with conditionally independent signals, while this difference is not found without the prespecified trading sequence.

It was assumed in[18;53] that the agents are myopic, however strategic agents will not always behave myopically as they may try to manipulate the beliefs of the other agents, and therefore, the market price in order to make extra profit. Conitzer[22] explored strategic aspects of prediction markets through their analogy to mechanism design. He observed that market scoring rules only incentivize agents to report their beliefs truthfully if they are myopic, but with non-myopic agents truthful revelation cannot be guaranteed. He proposed that mechanism design was an appropriate technique to solve this problem since direct-revelation mechanisms that are incentive-compatible to agents do not give agents any incentive to misreport their private information (beliefs). The author acknowledged that direct-revelation prediction markets are not very practical, but argued for the need of a mechanism design-based prediction market model. He proposed a sample mechanism comprising of information agents, that have relevant information about the event, but not necessarily able to convert this information into a probability, and, probability agents, that do not necessarily have any information about the event, but are able to convert any given information to a probability. The proposed mechanism uses information agents to compute conditional probability of an event happening and then it uses the probability agents to get estimates of these conditional probabilities using standard prediction market.

2.4.2 Risk Behavior

Dimitrov *et al.*[28] tackled the second question: What happens when agents are not risk-neutral? In practice, most people are better modeled as being risk-averse in their decision making. They modeled agents as expected-utility maximizers with an arbitrary weakly monotone and concave utility function that captures their risk aversion. The authors focused on a general setting in which agents have unknown risk aversion, and study whether it is possible to guarantee myopic truthfulness while preserving other desirable properties of prediction markets. They specified that any prediction market-like mechanism should satisfy the following properties:

- *myopic strategyproofness*, giving agents no incentive to report untruthfully,
- *sequential trade*, giving agents the opportunity to update beliefs,
- a variant of *sybilproofness*, capturing the idea that trading under multiple identities does not yield any direct advantage
- *boundedness* of the expected subsidy.

They proposed one mechanism that satisfies all of these properties, even with the agents that have unknown risk-averse preferences. The key building block of their result is a technique of scoring agents by varying their probability of winning a fixed reward. However, the mechanism has the undesirable property that expected profits decrease exponentially as the number of agents grows, and moreover, they proved that this is unavoidable in the context of mechanisms for arbitrarily risk-averse agents.

Iyer *et al.*[52] studied a setting where there are n risk-averse agents with conditionally independent private signals that participate in the prediction market with an automated market maker. They identified a condition called *smoothness requirement* under which a prediction market will aggregate the private information of rational risk-averse agents. The smoothness condition requires that there is no bid-ask spread for buying or selling a small quantity of any security. They showed that if the securities held by the agents converged over time, and the limiting market price charged by the market maker is continuously differentiable at zero with respect to the quantity traded, then the prediction market will aggregate information at any Perfect Bayesian Equilibrium (PBE). The authors proved that if the signal space is finite, then in any pure strategy PBE that satisfies asymptotic smoothness and bounded loss, prediction market aggregates information of the agents. The authors found that the market prices in general are not equal to the posterior state distribution because market prices must also reflect the risk aversion of the agents, i.e. the marginal expected utility of the agents. But they showed that when there is at least one risk-neutral agent in the prediction market along with risk-averse agents, the market prices are equal to the posterior probability. Although, the authors characterized conditions for full information aggregation at PBEs, the existence of such equilibria is still an open question.

2.5 Prediction Market for Decision Making

The work surveyed in this chapter so far has implicitly assumed that trading agents can not take actions to influence the outcome of the event. This is often not true in the real world. For example, with an internal prediction market to predict a software delivery date, a developer of the software who purchases securities in the prediction market may deliberately take actions outside of the prediction market to affect the software delivery date so that the securities he buys will pay off. Hanson[46] and Berg and Rietz[5] first hinted on the idea that prediction markets can be used as decision support systems, calling such markets *decision markets*. Once incorporated into the decision making process, prediction markets often unintentionally create incentives for trading agents to manipulate the market price. An emerging line of research has studied incentive issues that arise when using prediction markets for decision making.

Othman and Sandholm[84] looked at information elicitation from one agent by the decision maker. In their setting the decision maker chooses an action from a set of possible actions, $D = \{d_1, ..., d_k\}$, in order to maximize the probability of achieving a desirable outcome o_i, which could for example be "product launch by the beginning on 2013". The decision maker asks an agent to tell him the probability of achieving the outcome o_i under each of the alternative actions. Based on the agent's report, the decision maker makes his decision. The agent is risk-neutral and does not care which action is taken or whether the outcome o_i is achieved, it just cares about its payoff and how its payoff gets affected. To encourage an agent to submit an accurate probability, the decision maker pays the agent using a scoring rule.

The authors showed that there does not exist any strictly proper scoring rule and decision rule pair. They then created an asymmetric quasi-strictly proper rule for the *max* decision rule, where *max* decision rule is the rule that allows the decision maker to select the action that has the highest reported probability of achieving the desired outcome and where ties are broken in some fixed way. The rule is called a *quasi-strictly proper rule* if, given agent's true belief $p = (p_1, ..., p_k)$, d_j - the action taken by the deterministic decision rule D, and $u(p)$ - the agent's expected utility from reporting p given the decision rule/scoring rule pair, $u(p) \geq u(p'), u(p) > u(p^j)$, $\forall p' \neq p$ and where p^j represents any report for which the j^{th} component does not equal p_j (that is, the agent does not report $P(o_i|d_j)$ truthfully). The authors have shown that in a decision making prediction market with multiple agents, the agents always have an incentive to manipulate using the standard scoring rules. They proposed a family of new scoring rules that reduces the possibility of manipulation.

Chen and Kash[12] expanded Othman and Sandholm's work in[84]. They used the same model but they allowed randomized decision rules and considered multiple possible outcomes, instead of just two outcomes. Their main theorem provides a simple test to determine whether the scoring rule is proper for an arbitrary decision rule and scoring rule pair and generalizes result in[84] by characterizing all (strictly) proper scoring rules for all decision rules. The authors also discussed how the elicitation problem becomes more complicated when there are multiple agents; while strictly proper scoring rules can be used for a single agent and extended to prediction markets with multiple agents, these scoring and decision rule pairs do not have a such a natural extension. In particular, the authors identified two main problems. First of all, an agent has to base its decisions on beliefs about what the final market probabilities would be since only one decision is made in the end. Secondly, scoring rules that encourage truthful revelation no longer have the same effect in this situation since no individual agent controls the final decision.

Thus to combat these problems, in[19] the authors characterized all scoring rules incentivizing a single risk-neutral agent to report truthfully given a decision rule. They introduced myopic incentive-compatible decision markets. Unlike previously proposed models of decision markets, the authors calculated the payoff for agents using a decision scoring rule instead of a standard scoring rule. They found that when a decision maker risks taking an action at random, these decision scoring rules allow the rewards of unlikely actions to be increased and the rewards of likely actions are comparatively reduced, making risk-neutral agents indifferent to their effect on the decision. The authors also found that the risk of taking an action at random is necessary for myopic incentive-compatible decision markets, and reducing this risk increases the decision maker's worst-case loss.

Shi, Conitzer, and Guo[99] examined settings where agents participating in a prediction market may also have an ability to influence the outcome. Their setting comprises of a principal (i.e. the company) that sets up a prediction market and has a preference over the event outcomes (i.e. on-time delivery of the software), and, a group of agents that has information about the event of interest and can take actions to affect the outcome. The authors characterized all *principal-aligned* proper scoring rules that do not incentivize agent to take any action that may harm the principal in expectation. They showed an overpayment result, which roughly states that with n agents, any prediction mechanism that is principal-aligned will, in the worst case, require the principal to pay $\Theta(n)$ times as much as a prediction market. However, in this work, unlike works in[12;84], the elicited information is not explicitly used for decision making.

Boutilier[9] continued the study of realistic settings, in which the decision maker takes an action based on the agent's belief and the agent has an inherent interest in the decision, by presenting a formal model of scoring rules, in the form of compensation functions, that incentivize truthful reporting even when agents have an interest in the actions taken by the decision maker. Unlike any of the previous works, Boutilier explicitly modeled the decision maker's policy, utility, or loss in case of manipulation, and considered agent's utility apart from the payoff offered by the market maker. By analyzing a single agent setting, Boutilier found that the agent does not need to know the decision maker's policy prior to reporting its probabilities as long as it can verify which action has been taken after the fact and that the decision maker cannot ensure truthful reporting without full knowledge of the agent's utility function. However, in a multiple agent setting, he noted that more development needs to be done to make compensation rules applicable to and practical in prediction markets.

The incentives for the trading agents to manipulate the market price in a prediction market used for decision making can take the form of the potential to profit in a subsequent market. Dimitrov and Sami[27] examined incentive problems when there are two prediction markets for two different but related events. They showed that this situation might cause an agent to report sub-optimally in one market to mislead the agent in another market, and, that the information revealed and agent's payoffs in equilibrium are uniquely determined, and consistent with a minimax strategy profile. Chen *et al.*[20] extended the work in[27] with the outside incentives being of a general form of any monotone function of the final market price instead of the form of the potential profit in a second market used in[27]. They found that, for the case when the first agent has the outside payoff with probability 1, i.e. when the existence of the outside incentives to manipulate the market price is certain and common knowledge, in many cases there exists a separating equilibrium for the market where information is fully aggregated. They also showed that, for case when the probability for the first agent to have the outside payoff is less than 1, i.e. the existence of outside incentives is uncertain, there exists no separating or semi-separating equilibrium where information is fully aggregated if the outside incentive is sufficiently large.

2.6 Application Domains of Prediction Markets

Up to now, we have reviewed various mathematical and computational models for prediction markets where the main objective is to aggregate the traders' information to forecast the outcome of a future event or to make decisions. In contrast, some recent work has applied prediction markets to various domains beyond event outcome forecasting and decision making. In[31], a Turing Trade was created, which was a Web-based game that was a hybrid of a prediction market and a Turing test. Turing Trade allowed the users to buy securities (place bets) on whether they are talking to a human or a computer. The results were then analyzed quantitatively to determine how human-like a particular bot is. In a Turing Trade, a group of agents conversed with a single target. Each individual agent in the group could ask the target public questions, and the target gave public answers. During the conversation, all agents in the group participated in the prediction market, where the single binary event that the judges were trying to predict was "the target will be revealed to be a human." They were doing this by buying and selling securities using points, not real money. At the end of the game, the target's true nature (human or computer) was revealed, and based on this outcome some of the securities paid out. About 900 games of Turing Trade were played and the authors were able to collect large quantities of fine-grained data and to introduce a novel, fast-paced prediction market. By analyzing the data collected from the Turing Trade games, the authors found that Turing Trade produced very strong and accurate predictions after short periods of time, and that the market price responded rapidly to good or bad answers by the target.

Ellis and Sami[30] used a prediction market to supplement classroom learning. The authors carried out a quasi-experiment in an introductory political science class at the University of Michigan, Ann Arbor by creating 12 prediction markets with MSR market makers that were relevant to the topic of the course. The purpose of the study was to analyze the effect of prediction markets on student engagement with the course material. After analyzing their data, the authors concluded that an elective upper-level undergraduate course or a graduate course may be more appropriate settings for using prediction markets as an educational tool.

Pfeiffer and Almenberg[93] discussed the benefits of using prediction markets for biomedical research through an example application in the context of decision making in research on the genetics of diseases. When researchers investigate gene-disease associations the case-control studies are very useful, where a case-control study a group of patients that have been diagnosed with a disease is matched with a healthy control group, and the researchers then estimate whether a particular genetic marker is more frequent in one of the two groups.

The authors argue that prediction markets could be used to obtain prior probabilities for the association between genetic markers and the disease. The authors also discussed that prediction markets have a potential to improve the performance of not only biomedical research, but a general scientific research as well, by examining the ways to link prediction markets with scientific publishing.

Barbu and Lay[4] introduced a mathematical theory of simulating prediction markets numerically for the purpose of supervised learning of probability estimators. They presented a novel method for fusing the prediction information of features or trained classifiers, where the fusion result is the market price of the securities corresponding to the possible outcomes. They have shown that their obtained artificial prediction market is a maximum likelihood estimator and that artificial prediction market generalizes linear aggregation, logistic regression and even some kernel methods. The authors conducted experiments, where they compared the artificial prediction markets with other estimators, and their results showed that the artificial prediction markets often outperform random forest and implicit online learning on synthetic data and real datasets.

In the following chapters we present our contributions in solving open problems related to different aspects of prediction markets.

Chapter 3

Analyzing the Effect of Information on Prediction Markets

Information about an event that the trading agents receive affects their belief values about the outcome of an event, influences the prices corresponding to the event and finally determines the outcome of the event. Therefore, it makes sense to analyze the behavior of the trading agents in response to different information-related parameters in a prediction market. We develop a multi-agent based system that incorporates different information-related aspects including the arrival rate of the information, the reliability of the information, the accessibility of the information among the different trades and the perception of impact of the information by the trading agents. The multi-agent implementation of a prediction market allows us to easily analyze and verify the trading agents' behavior while varying different market and agent related internal parameters of the prediction market, as well as external parameters related to the information about events arriving at the market. To build our multi-agent system, we use modeling parameters obtained from various sources such as existing analytical models of financial markets, empirical evidence and data from real prediction markets, and agent utility and belief theory. The research question that we are trying to answer in this contribution is reproduced below in Table 3.1. We perform extensive simulations of our agent-based prediction market for analyzing the effect of information related parameters on the trading agents' behaviors expressed through their trading prices. We also compare our prediction market's behavior with an existing prediction market model, and, our agents' strategies with the zero-intelligence(ZI) agent strategy that has been formerly used for strategic pricing in prediction markets. The results show that our agent-based prediction market operates correctly and that our agents price predictions result in higher utilities than ZI agents. Next, we describe modeling of some of the information parameters we have used in our prediction market and some of our simulation results. More details of our modeling parameters and complete simulation results can be found in[55].

Contr.	Research Question	Research Approach
1	How do changes in different aspects of information affect the behavior of prediction markets?	Empirical Analysis

Table 3.1: Research question that is addressed in contribution 1 of this thesis.

3.1 Preliminaries

In this chapter we use a multi-agent prediction market to empirically analyze its behavior and the behavior of traders in response to different market parameters. In our prediction market, each human trading agent is modeled as a software agent called a *trading agent* that embodies the behavior of a human trading agent. We use a prediction market model that is similar to the continuous double auction(CDA) protocol. Each event e is associated with two prices in the market - a *bid* price $\pi^{buy}_{e,d}$ at which trading agents can purchase a security of the event and a *ask* price $\pi^{sell}_{e,d}$ at which trading agents can sell a security of the event during day d. Both of these prices are normalized to the range of $[0, 1]$ and trading agents calculate the current market price of an event e during day d, $\pi_{e,d}$, as the average of the current bid and ask prices of the event during the day. That is, $\pi_{e,d} = 0.5 \times (\pi^{buy}_{e,d} + \pi^{sell}_{e,d})$.

trading agents then interpret the current market price $\pi_{e,d}$ for the event e on day d as the expected probability of the event e happening (that is, its outcome being 1).

A trading agent $n \in N$ observes the information set $\mathcal{I}_{n,e,d}$ associated with event e on day d and decides to place an order to buy the security associated with event e at a price $\pi^{buy}_{n,e,d}$ or to sell the security associated with event e at a price $\pi^{sell}_{n,e,d}$. A *market order* for trading a security contains the identifier and number of units of the security the trading agent wishes to buy or sell. Market orders are executed instantly in the market, but the price of $\pi^{buy}_{n,e,d}$ (for purchase orders) or $\pi^{sell}_{n,e,d}$ (for sell orders) is not guaranteed to trading agent n for buying or selling the security. Instead, if a trading agent wishes to purchase the security, it has to pay a price of $\pi^{buy}_{e,d}$, corresponding to the maximum price that has been offered by any trading agent for purchasing a security related to the event during the day d. $\pi^{buy}_{e,d}$ is called the *bid (buy) price* of the event and is given by $\pi^{buy}_{e,d} = \max_{\mathrm{n \in N}} \pi^{buy}_{n,e,d}$. Similarly, when a trading agent wishes to sell the security, it has to pay a price of $\pi^{sell}_{e,d}$, where $\pi^{sell}_{e,d} = \min_{\mathrm{n \in N}} \pi^{sell}_{n,e,d}$. $\pi^{sell}_{e,d}$ is called the *ask (sell) price* of the event and corresponds to the minimum price that a trading agent has offered for selling a security related to the event during day d.

3.2 Agent-based Prediction Market

3.2.1 Trading Agents: Functionality and Parameters

One of the most important parameters in a prediction market is the perception of the probability of occurrence of an event by the human traders. This parameter affects the dynamic values of the prices of securities corresponding to the different events in the market as well as the quantities of different securities each trader buys or sells. To model the perception of event occurrence probabilities, each trading agent n in our prediction market uses a private belief $b_{n,e} \in [0, 1]$ that corresponds to the probability that the event e will occur with an outcome = 1. Existing models of prediction markets[34;69;82] have empirically verified that this belief value is drawn from a Beta distribution. Following these studies, we have assumed in our prediction market that a trading agent n's initial belief about the outcome of an event being 1 is drawn independently from a continuous beta distribution $Beta(\alpha, \beta)$, where α and β are positive shape parameters that determine the shape and skew of the beta distribution's curve. There are two cases when the belief of an agent is dynamically updated. The first case is when the market price changes and the second case is when a new information about the event becomes available to the trading agent. For the first case, the belief update function used by our trading agents is based on Gjerstad's prediction market model[34] where, during each day, every time a market price of the security for an event e changes, trading agent n updates its beliefs as a weighted sum of the event e's observed market prices during that day and its own beliefs using the belief update Equation 3.1.

$$b_{n,e} \leftarrow \mu_{n,e,d} \cdot b_{n,e} + (1 - \mu_{n,e,d}) \cdot \pi_{e,d}, \tag{3.1}$$

where $\pi_{e,d}$ denotes the current market price of the security for event e during day d, $b_{n,e}$ is the belief of agent n for event e, and $\mu_{n,e,d} \in (0, 1)$ is agent n's belief weight factor[111] that represents the confidence of agent n in its current belief value about the outcome of an event e. At the end of each day, agent n finds out whether its buy or sell order for that day was executed or not and updates its belief weight factor using the equation (with U representing a uniform distribution):

$$\mu_{n,e,d} = \begin{cases} U[0.5, 1] & \text{, if buy/sell order executed;} \\ U[0, 0.5] & \text{, if buy/sell order wasn't executed.} \end{cases}$$

This ensures that if the buy/sell order was executed (not executed) an agent gives a higher (lower) preference to its own previous belief value while updating its belief using Equation 3.1. The second case of the belief update is discussed in Section 3.2.3 after we introduce the information modeling in our prediction market.

3.2.2 Dynamic Information Modeling in Prediction Markets

Empirical data from real prediction markets such as the Iowa Electronic Marketplace show that the belief value of trading agents for an event e is significantly affected by new information that arrives in the market. However, previous prediction market models do not consider the effect of new event information on the belief values of a trading agent. To address this deficit, we have introduced a new belief update formula that uses new parameters in our prediction market, that are described below:

a. *Information impact parameter* $\zeta_{i,e}$: In a prediction market, the price of the security related to an event is changed by the traders every time new information about the event becomes available. Also, different information about an event result in different degrees of price change of the security. We incorporate the effect of new information on the prices in the prediction market using an information impact parameter $\zeta_{i,e} \in [-0.5, 0.5]$ for information i related to event e. Larger values of $\zeta_{i,e}$ indicate high-impact information or unexpected news related to an event, which significantly affects the price of security related to the event. Smaller values of $\zeta_{i,e}$ indicate a normal-impact or low-impact information. These correspond to expected news about an event and don't affect prices significantly.

b. *Information reliability parameter* $r_{n,e}$: Previous research on online financial markets[65] have shown that different people rely on the new information in the market to different degrees and that an information reliability parameter can be used to represent the level of agent n's reliance on information regarding event e. Following a study on the relation between trust and risk reported by Molm *et al.*[75], we consider that the risk coefficient of an agent, $\theta_{n,e,d}$, is affected by the reliability it associates with new information related to events. To model this, we introduce an information reliability parameter, $r_{n,e}$ of an agent n for an event e. If a trading agent n is a risk-taker ($\theta_{n,e,d} < 0$), then its reliability parameter is high ($r_{n,e} \in U[0.5, 1.0]$), if it is risk-neutral ($\theta_{n,e,d} = 0$), then $r_{n,e} = 0.5$, and if it is risk-averse ($\theta_{n,e,d} > 0$), then its reliability parameter is low ($r_{n,e} \in U[0, 0.5)$). Thus, new information about an event

e is perceived by the trading agents with different amounts of information reliability. To model this behavior, the trading agents in our prediction market dynamically update the reliability parameter at the end of each day using the following equation:

$$r_{n,e,d} = \begin{cases} U[0, 0.5] & , \text{if } \theta_{n,e,d} > 0; \\ 0.5 & , \text{if } \theta_{n,e,d} = 0; \\ U[0.5, 1.0] & , \text{if } \theta_{n,e,d} < 0. \end{cases}$$

c. *Information availability parameter* $a_{n,e}$: In real prediction markets, different traders have access to different amounts of information related to events. Traders with access to a larger amount of information related to events are able to make informed decisions about the price of the securities related to events. We model different levels of availability of information among the trading agents using an information availability parameter $a_{n,e} \in \{0, 0.5, 1\}$. $a_{n,e} = 1$ indicates that agent n has access to all available information about event e in the market, while $a_{n,e} = 0.5$ and $a_{n,e} = 0$ indicate limited and no access to information about event e respectively. Although the availability of information to human traders follows a more complex model, we restrict the information availability parameter to three discrete values to get easily analyzable results of the effect of information availability on the trading agents, without loss of generality.

3.2.3 Agent Belief Revision based on new information

On each day in the prediction market, a trading agent has to make a decision about the securities it wants to purchase, sell or hold. The belief value about the outcome of the events related to the securities plays a pivotal role in making this decision. Therefore, every agent has to update its belief values for each event it is interested in based on the new information in the market related to those events. Because new information about events can arrive into the market from various sources with different reliabilities, an agent should adjust the impact of the information with the corresponding information reliability parameter. The reliability-weighted information impact parameter of an event then increases the belief value about the outcome of that event if its positive, and decreases the belief value if it is negative. Finally, to prevent rapid changes to the belief value, belief updates should also include the current belief value of an event. The belief update equation based on these factors that is used by a trading agent in our prediction market is given by:

$$b_{n,e} \leftarrow r_{n,e,d} \cdot (b_{n,e} - \bar{\zeta_{i,e}} \cdot ln\ b_{n,e}) + (1 - r_{n,e,d}) \cdot b_{n,e} \tag{3.2}$$

where $\bar{\zeta}_{i,e} = \sum_i \zeta_{i,e}$ denotes the average value of the information impact parameter for all information related to event e by agent n, and $r_{n,e,d}$ denotes the information reliability parameter agent n has for event e on day d. A positive (negative) value of the information impact factor, increases (decreases) the new belief value from its previous value, while a zero value of the information impact factor does not change the belief value. Also, the agent with higher information reliability accounts more for the information impact, while the agent with smaller information reliability places more weight on its past belief.

3.2.4 Utility Maximization and Optimum Trading Quantity Calculation

The main problem facing a trading agent in a prediction market is to determine what quantities of each security to buy and sell on each day at the trading price of the security so that it can maximize its own utility or monetary gain. To achieve this, a trading agent dynamically incorporates new information about the event related to each security and dynamically obtains updates of the current market price of each security to estimate the optimum quantity of the security it should buy or sell. When the prediction market starts on day d_0, each trading agent n is provided with an initial wealth of $w_{n,d_0} > 0$. We consider each trading agent in our prediction market as a utility maximizer that buys or sells securities to maximize its wealth on each day. The instantaneous utility of agent n on day $d-1$ is updated by the following equation:

$$\acute{u}_{n,d} = \sum_{e \in E} q^{sell}_{e,d} \cdot \pi^{sell}_{e,d} - \sum_{e \in E} q^{buy}_{e,d} \cdot \pi^{buy}_{e,d} \tag{3.3}$$

where, $q^{sell}_{e,d}$ is the number of securities related to an event e sold by agent n on day d, $q^{buy}_{e,d}$ is the number of securities related to an event e bought by agent n on day d, and $\pi^{sell}_{e,d}$ and $\pi^{buy}_{e,d}$ are respectively the ask and bid price of the security for event e during day d.

Equation 3.3 maintains the instantaneous utility of the agent n on day d depending on the transactions it has made on that day, however it does not account for its risk type. Previous research[34;111] has shown that the correct behavior of prediction market models can be obtained only if the risk-taking and risk-averse behaviors of the human traders are considered while trading securities in prediction markets. Therefore, in our prediction market we adopt a constant relative risk averse (CRRA) utility function $\widetilde{u}_{n,d}$ for agent n with a relative risk aversion coefficient. CRRA utility functions have been widely used to model risk behaviors. Relative risk aversion coefficient, θ_n, is used to classify trading agent n's risk levels as follows. If $\theta_n > 0$, the agent n is risk-averse, if $\theta_n = 0$, the agent n is risk-neutral, and if $\theta_n < 0$, the agent n is risk-seeking. The trading agents' risk coefficients are normally distributed in our simulations. Following Gjerstad's trading agent utility model[34], during each day a trading agent uses its instantaneous utility and its risk-taking coefficient to calculate its modified instantaneous utility for that day, using Equation 3.4.

$$\widetilde{u}_{n,d}(\acute{u}_{n,d}, \theta_n) = \begin{cases} \frac{\acute{u}^{1-\theta_n}_{n,d}}{1-\theta_n} & , \text{if } \theta_n \neq 1; \\ ln(\acute{u}_{n,d}) & , \text{if } \theta_n = 1. \end{cases} \tag{3.4}$$

The utility value for an agent can fluctuate considerably in successive days based on its trading pattern, and the price of securities in the market. Large fluctuations in the utility can in turn delay the convergence to equilibrium because it can cause large fluctuations in the number of securities being bought or sold in the market. To prevent these fluctuations we have used a daily weighted utility value for an agent's utility given by:

$$u_{n,d}(\widetilde{u}_{n,d}) = \sum_{h=d}^{d-H} \delta_h \cdot \widetilde{u}_{n,d}, \tag{3.5}$$

where δ_h is a discount factor that ensures that more recent utility levels have a greater influence on the current weighted utility, $\sum_{h=d}^{d-H} \delta_h = 1$, and H is the number of days considered from the past.

The final objective of a rational trading agent is to maximize the utility it receives from purchasing and selling securities in the prediction market. To achieve this objective, a trading agent has to calculate the optimum quantity of securities it should trade so that the resulting utility is maximized. Let $Q_{n,e,d}$ denote the optimum quantity of securities related to event e for trading agent n during day d. The trading agent's decision is to choose quantity to buy or sell for each security, so that the expected utility is maximized. We have used a similar model in this part of the thesis with the expected utility given by equation below:

$$E_e(u_{n,d}(\theta_n, Q_{n,e,d})) = \begin{cases} \frac{1}{1-\theta_n} b_{n,e,d}(u_{n,d} + Q_{n,e,d} \cdot (1-\pi_{e,d}))^{1-\theta_n} + \\ (1-b_{n,e,d})(u_{n,d} - Q_{n,e,d} \cdot \pi_{e,d})^{1-\theta_n} & \text{, if } \theta_n \neq 1; \\ b_{n,e,d} \cdot ln(u_{n,d} + Q_{n,e,d} \cdot (1-\pi_{e,d})) + \\ (1-b_{n,e,d}) ln(u_{n,d} - Q_{n,e,d} \cdot \pi_{e,d}) & \text{, if } \theta_n = 1. \end{cases}$$

where $\pi_{e,d}$ is the current market price of the security for an event e on day d. The optimum quantity $Q_{n,e,d}$ is obtained by taking the first-order derivative of the expected utility equation above, setting the resulting equation equal to 0, and solving it to obtain the following solution:

$$Q_{n,e,d}(\pi_{e,d}, b_{n,e,d}, \theta_n, u_{n,d}) = \frac{((1-\pi_{e,d})^{\frac{1}{\theta_n}} b_{n,e,d}^{\frac{1}{\theta_n}} - (\pi_{e,d})^{\frac{1}{\theta_n}} (1-b_{n,e,d})^{\frac{1}{\theta_n}}) u_{n,d}}{((1-\pi_{e,d})(\pi_{e,d})^{\frac{1}{\theta_n}} (1-b_{n,e,d})^{\frac{1}{\theta_n}} + \pi_{e,d}(1-\pi_{e,d})^{\frac{1}{\theta_n}} b_{n,e,d}^{\frac{1}{\theta_n}})} \tag{3.6}$$

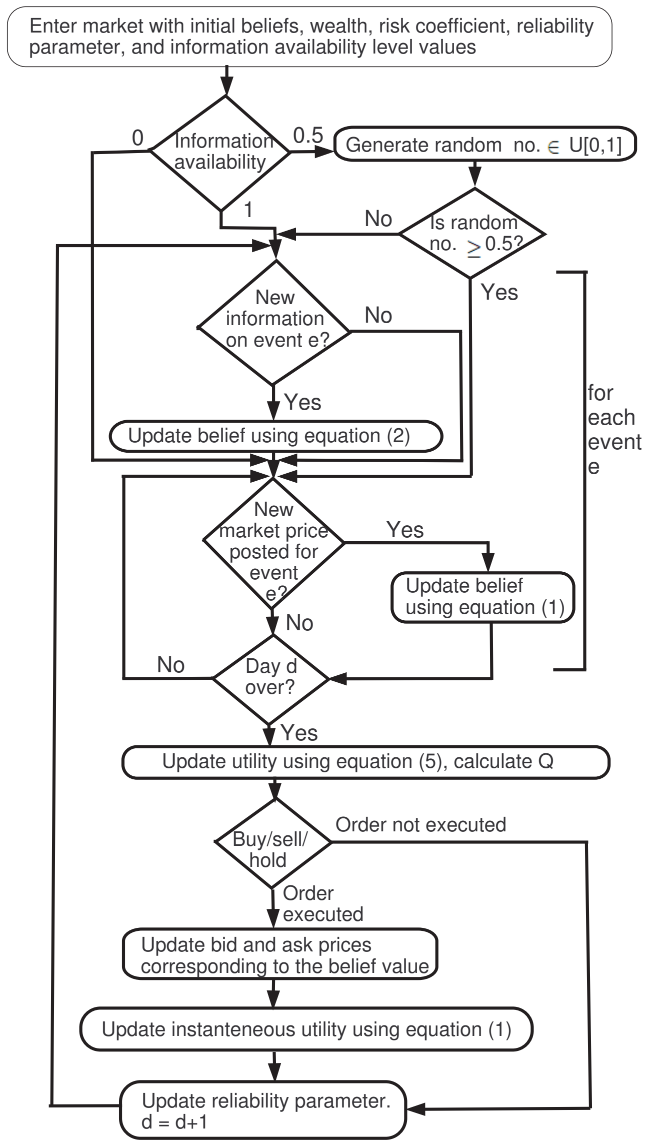


Figure 3.1: A flowchart showing the operation of the trading agents in the prediction market.

If $Q_{n,e,d} > q_{n,e,d}$, that is the quantity required to maximize utility on day d is more than the number of securities of event e that agent n currently has on day d, the agent places a buy order for purchasing $Q_{n,e,d} - q_{n,e,d}$ additional units of the security of event e. On the other hand, if $Q_{n,e,d} < q_{n,e,d}$, the agent places a sell order for selling $Q_{n,e,d} - q_{n,e,d}$ units of the security of event e. The agents set their prices corresponding to their current beliefs; that is, $\pi^{buy}_{n,e,d} = b_{n,e,d}$ if the trader is purchasing securities of event e and $\pi^{sell}_{n,e,d} = b_{n,e,d}$ if the trader is selling securities of event e during day d.

Figure 3.1 gives a flowchart of the algorithm used by a trading agent in our agent-based prediction market. Each trading agent n enters the prediction market with some initial wealth, a belief about the outcome of each event that is initially drawn from a Beta distribution, a risk coefficient, and, information reliability and information availability parameter values. First, the agent checks its information availability parameter $a_{n,e}$ and updates its beliefs about an event e (using Equation 3.2) with a probability corresponding to the value of $a_{n,e}$. Throughout the day, the trading agent n checks whether the market price for the security associated with an event e has been updated. When the market price is changed, the trading agent updates its beliefs using Equation 3.1. At the end of each day, the trading agent n updates its utility using Equation 3.5 and calculates $Q_{e,d}$, the number of units of security corresponding to event e it needs to maximize its utility. Based on $Q_{e,d}$ value it then makes a decision to buy, sell, or to hold a security associated with an event e and submits an order to the prediction market. If the order is accepted, the trading agent n updates its weighted utility using Equation 3.3. Finally, the trading agent's information reliability parameter is updated and the new day starts.

3.3 Simulation Results

We verify the performance of our prediction market through several simulations while varying different market, event and agent related parameters. The first objective of our simulations is to clarify that our prediction market behaves in a manner similar to actual prediction markets with human traders. Secondly, we attempt to understand the dynamic behavior of trading agents' prices and utilities under different values of the information related parameters.

3.3.1 Model Comparison

For our first set of experiments, we compare the performance of our multi-agent prediction market with the model proposed by Gjerstad[34]. The price data for this experiment was taken from the 2008 Democratic Presidential Nomination Market in IEM[71]. There are four securities in this market representing four different outcomes, i.e., Hillary Clinton, John Edwards, Barack Obama, or another candidate winning the democratic presidential nomination. Figure 3.2 shows the difference between the actual market price and the pre-

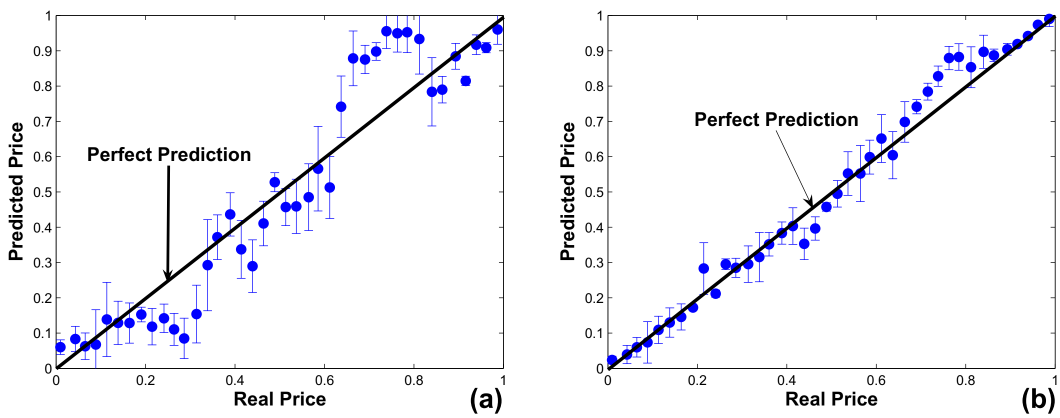


Figure 3.2: The comparison of the simulated market prices with real market prices in a prediction market based on Gjerstad's model(a) and our agent-based prediction market(b). The solid line represents perfect prediction of market prices by agents.

dicted market price while using our agent-based prediction market (a) and the prediction market based on the Gjerstad's model (b) for the security corresponding to Obama winning the nomination. The error bars show the confidence intervals over 10 runs. As shown in Figure 3.2, the difference between the predicted and actual prices using our agent-based prediction market is significantly lower (8%) than that reported in Gjerstad's model (35%). The more accurate price prediction in our prediction market can be attributed to our belief update mechanism that considers two additional parameters - the risk coefficient of each agent through the information reliability parameter $r_{n,e}$, as well as the information impact parameter $\zeta_{i,e}$, as compared to Gjerstad's model. These parameters incorporate the information flow across the agents in the market as well as the effect of the information of the agents, and, therefore, enable the agents to make a more informed decision about the belief related to the event in the market.

For all of our other simulations in this contribution, we use price data from the movie markets in Iowa Electronic Marketplace(IEM)[71]. A security in a movie market on IEM

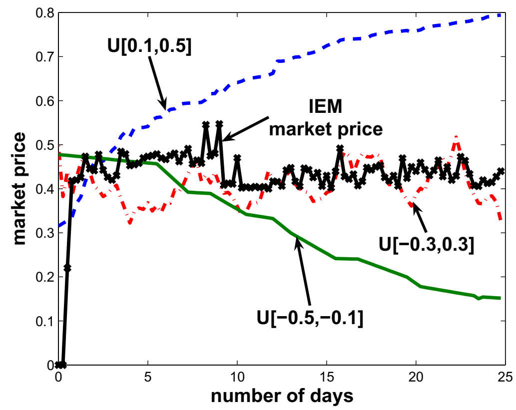


Figure 3.3: A graph showing the effect of the information impact parameter on the market prices.

corresponds to a boolean event - whether the box office collections of a certain Hollywood movie will reach a predetermined dollar amount. The movie market securities are particularly suitable for our agent-based prediction market because like the events in our market, the events corresponding to the different movies are not correlated with each other.

3.3.2 Effect of information impact

In a prediction market, the same information about an event can cause different traders to behave differently based on the traders' perceptions of the impact of that information. In our next set of experiments, we attempt to analyze the effect of different traders having different information impact parameters on the belief values of agents in the market. Our definition of the dependency between agent belief values and the information impact parameters is given in Equation 3.2, which is reproduced below:

$$b_{n,e,d} = r_{n,e,d} \cdot (b_{n,e,d-1} - \bar{\zeta_{i,e}} \cdot ln\ b_{n,e,d-1}) + (1 - r_{n,e,d}) \cdot b_{n,e,d-1}$$

From this equation, we see that a positive(negative) value of the information impact factor causes an increase(decrease) in the belief value of an agent for a security, while a zero value of the impact factor does not change the belief value. This increases the utility the agent receives and results in more buy(sell) orders from the agent (using Equation 3.6) before the event expires. To verify the effect of different values of the information impact factor, we consider the effect of a positive-only impact factor ($\zeta_{i,e} \in U[0.1, 0.5]$), a negative-only impact factor ($\zeta_{i,e} \in U[-0.5, -0.1]$), and the impact factor consisting of mixed positive and negative values ($\zeta_{i,e} \in U[-0.3, 0.3]$).

In prediction markets, trading agents' beliefs about an event are expressed through the aggregated market price of the event. We have therefore used the real closing prices of the *MAT45H* security that has a duration of 25 days from the IEM movie data to analyze the effect of information impact factor on the agents' beliefs. For this set of experiments, each trading agent n is assumed to have fixed information reliability parameter $r_{n,e} = 0.5$ about each event e. In Figure 3.3, we report the results of the different values of the information impact parameter on the aggregate market prices of the agents over the 25-day duration of the security.

Figure 3.4 shows the means and the variances of the market prices corresponding to different values of the information impact factor from the results shown in Figure 3.3. The Pearson's correlation coefficient between the means of the information impact factor values and the means of the market prices obtained under the corresponding $\zeta_{i,e}$ values is 0.98. A positive value of the correlation coefficient close to 1 shows that market prices are strongly correlated with the value of the information impact parameter. When the impact factor of the new information about an event has mixed positive and negative values, then it does not have a significant impact on the market prices as shown by the $U[-0.3, 0.3]$ curve in Figure 3.3.

Mean of $\zeta_{i,e}$ distribution	Mean Price	Variance in Price
0.3	0.53	0.04
0	0.49	0.07
-0.3	0.45	0.05

Figure 3.4: Mean and variance of the market prices under different $\zeta_{i,e}$ values.

3.3.3 Effect of information availability

In this set of experiments we set out to test if different information availability levels related to events have any effect on trading agents' beliefs and the agents' utilities. We have once again used agents' ask prices to illustrate the relationship between the information availability and the agents' beliefs because agents express their beliefs by setting their bid and ask prices. By analyzing ask prices, we can also see how different the information availability values lead to different prices being set by agents, which in turn leads to different values in utility. Different levels of availability of information among the trading agents can result in different trading decisions made by the trading agents and therefore in different utility. For example, a trading agent with all the information about an event available with it, can benefit from that information by making more informed trading decisions and get a higher utility. We have used the information availability parameter described in Section 3.2.2, $a_{n,e} = \{0, 0.5, 1\}$, to denote different degrees of information penetration among the trading agents. We have again reported the results using the closing prices and event's duration from the *MINC180H* security of the IEM movie market. In our simulations, one third of the trading agent population has no access to the information about the event with $a_{n,e} = 0$, another third of the agent population has partial access to the information with $a_{n,e} = 0.5$, while the final third of the population has full access to the information with $a_{n,e} = 1$. The results of our simulations are shown in Figure 3.5.

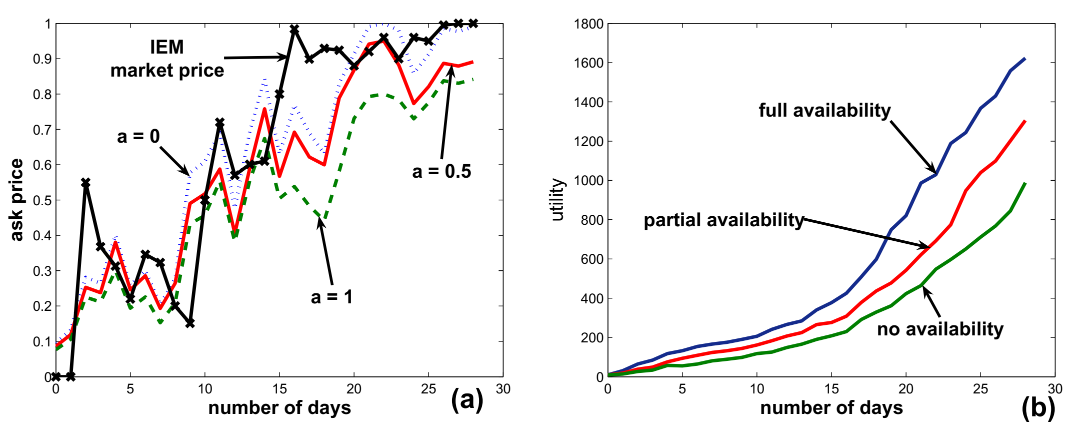


Figure 3.5: Effect of the different levels of information availability on (a) the trading agents' ask prices and (b) their utilities.

From these results, we calculated the Pearson's correlation coefficient between the information availability parameter and the market price and found it to be -0.88. This indicates that there is a strong negative association between the market price and the information availability parameter. This implies that the trading agents with more information (higher value of $a_{n,e}$) about the event are able to set the lowest ask prices for the event's security and purchase the security. Figure 3.5(b) shows that trading agents with full information availability are able to obtain higher utility than other agents. This behavior can be attributed to the fact that if $a_{n,e} = 1$, the agent's belief function update using Equation 3.2 is always triggered to include new information about the event e if available, as shown in the flowchart in Figure 3.1. This results in more up-to-date values of beliefs (using Equation 3.2) and ensures more accurate value of the quantity of security that the agent should trade (using Equation 3.6) and culminates in higher utility to the trading agent. On the other hand, if $a_{n,e} = 0.5$, the agent's belief function update using Equation 3.2 is triggered only 50% of the time, resulting in less utility for agent n. When $a_{n,e} = 0$, the agent n's belief function update using Equation 3.2 is never triggered, resulting in the least utility for agent n. Our result also agrees intuitively with the behavior of real traders in prediction markets - more information about an event enables a trader to make a more informed decision and thereby obtain higher utilities.

3.3.4 Effect of scalability

For our next set of experiments in this chapter, we have tested the scalability of our multi-agent based prediction market with the number of trading agents and the number of events. We have tested our prediction market with different combinations of $5, 10, 15, 20$ and 25 events, and, $50, 100, 300, 500$ and 1000 trading agents in the market. Table 3.2 shows the average utility for a trader at the end of 100 days. We observe that as the number of events in the market increases, the traders' utilities increase by 30% because they are able to trade in more securities. In a similar manner, when the number of trading agents in the market increases but the number of events is fixed, the average utility of a trader decreases by 40% because of the increased competition between traders in the market. In summary, as

Traders→/ Events↓	**50**	**100**	**300**	**500**	**1000**	Avg. over no. of agents
5	1480.5	1191.9	1071.0	1003.6	893.5	1059.1
10	1560.0	1265.0	1187.5	1089.6	965.2	1183.6
15	175716	1456.8	1299.2	1174.6	1006.5	1306.3
20	1791.1	1545.5	1424.5	1287.5	1104.4	1397.8
25	1843.3	1643.2	1534.5	1393.8	1275.2	1511.2
Avg. over no. of events	1631.5	1406.7	1295.0	1153.9	970.9	

Table 3.2: Utilities of one trader from the simulations with different number of traders and events.

the prediction market becomes more populated in the number of events and the number of agents, the utilities of the trading agents increase correspondingly.

Our simulation results quantify the effect of information on the price dynamics and utilities of trading agents in a prediction market. We briefly summarize our simulation results. From our results we observe that the trading agents with a higher value of the information reliability are able to obtain higher utilities. This allows us to conclude that a higher reliance of the trading agents on the information in the market allows them to get more news about the event and make more informed decisions. We also observe that the trading agents with more information (higher value of information availability parameter) about the event are able to obtain higher utilities. This result suggests that the trading agents that have access to more information are able to use the available information to obtain higher utilities. This also consistent with the behavior of human traders in real prediction markets, where well-informed traders are able to make better decisions. Overall, our results suggest that different aspects of the information about events in a prediction market have a significant impact of the prices, utilities and probabilities of the events in the market. This and other simulation results obtained for this part of the thesis can be used to obtain a better understanding of traders' behavior in a prediction market in response to information about events. The work presented in this chapter was published in[55] and the overview of our findings is shown in Table 3.3.

Contr.	Research Question	Research Findings
1	How do changes in different aspects of information affect the behavior of prediction markets?	- Higher information arrival rate leads to lower price fluctuations and higher utilities. - Higher impact information tends to increase the prices. - Higher reliability of information leads to more stable prices and higher utilities. - Higher information availability results in, lower prices and higher utilities.

Table 3.3: Research findings for the research question in contribution 1 of this thesis.

Chapter 4

A Multi-Agent Prediction Market based on Partially Observable Stochastic Games for Analyzing the Behavior of the Trading Agents

In this work, we address the problem of designing a formal and realistic framework of prediction market that can be used to model the interactions of the software trading agents. Our main goal is to develop a strategy for trading agents that when used in our proposed framework can improve information aggregation in a prediction market as well as improve the utilities of the trading agents in general.

The work in this chapter builds upon previous directions of research on prediction markets and develops a game-theoretic formal model for the strategic behavior of trading agents. The main contribution of this chapter is to use a partially observable stochastic game (POSG)[45] that can be used by each agent to reason about its actions. Within this POSG model, we calculate the correlated equilibrium strategy for each agent using the aggregated price from the market maker as a recommendation signal. We have also consider the risk preferences of trading agents in prediction markets and show that a Pareto optimal correlated equilibrium solution can give incentives for truthful revelation to risk averse agents. We then compare the POSG/correlated equilibrium based trading strategy with five different trading strategies used in similar markets with the data obtained from the Intrade prediction market events. Our results show that the agents using the correlated equilibrium strategy profile are able to predict prices that are closer to the actual prices that occurred in real prediction markets and these traders also obtain $35 - 127\%$ higher utilities. The research questions that we try to answer in this contribution are reproduced below in Table 4.1.

Contr.	Research Question	Research Approach
2	How do different trading agent behaviors affect the behavior of prediction markets?	Empirical Analysis
2	What trading strategies perform the best in prediction markets?	Empirical Analysis
2	How can prediction markets incentivize trading agents to participate in order to achieve a higher prediction accuracy?	Theoretical Examination
2	How does the trader's behavior using a formal game-theoretic model compare to the trader's behavior in real prediction markets?	Empirical Analysis

Table 4.1: Research questions that are addressed in contribution 2 of this thesis.

4.1 Partially Observable Stochastic Games for Trading Agent Interaction

As before, we assume that each human trader is represented by a software trading agent that buys and sells securities on behalf of the human trader. For simplicity of explanation, we consider a prediction market where a single security is being traded over a certain duration. This duration is divided into trading periods, with each trading period corresponding to a certain time period in a real prediction market. The 'state' of the market is expressed as the quantity of the purchased units of the security in the market. At the end of each trading period, each trading agent receives information about the state of the market from the market maker. With this prior information, the task of a trading agent is to determine a suitable quantity to trade for the next trading period, so that its utility is maximized. In this scenario, the environment of the agent is partially observable because other agents' actions and payoffs are not known directly, but available through their aggregated beliefs. Agents interact with each other in stages (trading periods), and in each stage the state of the market is determined stochastically based on the actions of the agents and the previous state. This scenario directly corresponds to the setting of a partially observable stochastic game[33;45]. A POSG model offers several attractive features such as structured behavior by the agents by using best response strategies, stability of the outcome based on equilibrium concepts, lookahead capability of the agent to plan their actions based on future expected outcomes, ability to represent the temporal characteristics of the interactions between the agents, and, enabling all computations locally on the agents so that the system is robust and scalable.

Previous research[55] has shown that information related parameters in a prediction market such as information availability, information reliability, information penetration, etc., have a considerable effect on the belief (price) estimation by trading agents. Based on these findings, we posit that a component to model the impact of information related to an event should be added to the POSG framework. With this feature in mind, we propose an interaction model called a partially observable stochastic game with information (POSGI) for capturing the strategic decision making by trading agents. A POSGI is defined as: $\Gamma = (N, S, (A_i)_{i\in N}, (R_i)_{i\in N}, T, (O_i)_{i\in N}, \Omega, (\mathcal{I}_i)_{i\in N})$, where N is a finite set of agents, S is a finite, non-empty set of states - each state corresponding to certain quantity of the security being held (purchased) by the trading agents. A_i is a finite non-empty action space of agent i s.t. $\overline{a_k} = (a_{1,k}, ..., a_{|N|,k})$ is the joint action of the agents and $a_{i,k}$ is the action that agent i takes in state $k \in S$, where agents take actions sequentially. In terms of the prediction market, a trading agent's action corresponds to certain quantity of security it buys or sells, while the joint action corresponds to changing the purchased quantity for a security and taking the market to a new state. $R_{i,k}$ is the reward or payoff for agent i in state k which is calculated using the LMSR market maker. $T : T(s, \overline{a}, s') = P(s'|s, \overline{a})$ is the transition probability of moving from state s to state s' after joint action $\overline{a}$ has been performed by the agents. O_i is a finite non-empty set of observations for agent i that consists of the market price and the information signal, and $o_{i,k} \in O_i$ is the observation agent i receives in state k. $\Omega : \Omega(s_k, I_{i,k}, o_{i,k}) = P(o_{i,k}|s_k, I_{i,k})$ is the observation probability for agent i of receiving observation $o_{i,k}$ in state s_k when the information signal is $I_{i,k}$. Finally, $\mathcal{I}_i$ is the information set received by agent i for an event $\mathcal{I}_i = \bigcup_k I_{i,k}$ where $I_{i,k} \in \{-1, 0, +1\}$ is the information received by agent i in state k. The complete information arriving to the market $\mathcal{I} = \bigcup_{i \in N} \mathcal{I}_i$ is temporally distributed over the duration of the event. Information that improves the probability of the positive outcome of the event is considered positive ($I_{i,k} = +1$) and vice-versa, while information that does not affect the probability is considered to have no effect ($I_{i,k} = 0$). For example, for a security related to the event "Obama wins 2008 presidential elections", information about Oprah Winfrey endorsing Obama would be considered high impact positive information and information about Obama losing the New Hampshire Primary would be considered negative information.

Based on the POSGI formulation of the prediction market, the interaction of an agent with the environment (prediction market) and the information source can be represented by the transition diagram shown in Figure 4.1*. The environment (prediction market) goes through a set of states $\tilde{S} = \{s_1, ..., s_H\} : \tilde{S} \in S$, where H is the duration of the event in the prediction market and s_h represents the state of the market during trading period h. This state of the market is not visible to any agent. Instead, each agent i has its own internal belief state $B_{i,h}$ corresponding to its belief about the actual state s_h. $B_{i,h}$ gives a probability distribution over the set of states S, where $B_{i,h} = (b_{1,h}, ..., b_{|S|,h})$. Consider trading period $h - 1$ when the agents perform the joint action $\overline{a}_{h-1}$. Because of this joint action of the agents the environment stochastically changes to a new state s_h, defined by the

*We only show one agent i to keep the diagram legible, but the same representation is valid for every agent in the prediction market. The dotted lines represent that the reward and environment state is determined by the joint action of all agents.

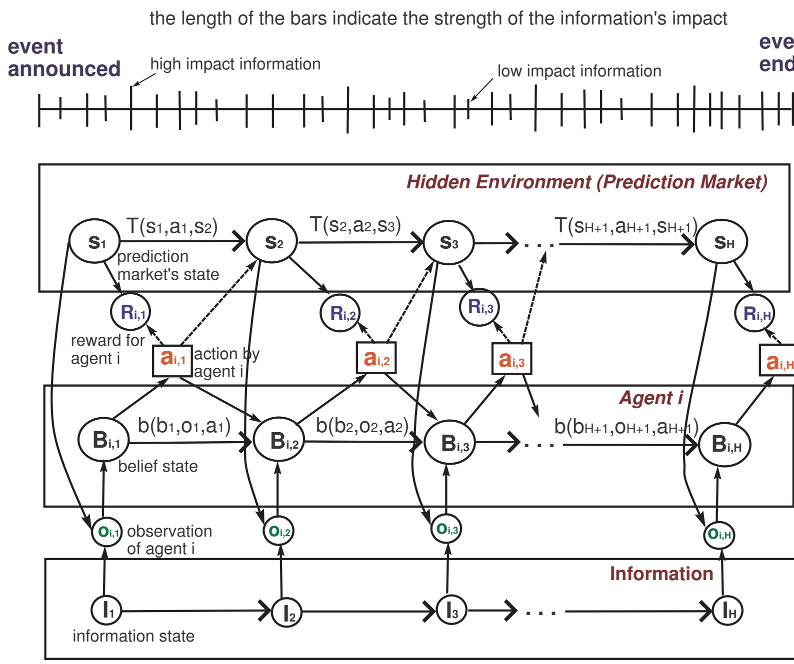


Figure 4.1: An agent interactions with the hidden environment (prediction market) and an external information source.

state transition function $T(s_{h-1}, a_{h-1}, s_h)$. The agent i doesn't directly see the environment state, but instead receives an observation $o_{i,S_h} = (\pi_{s_h}, \mathcal{I}_{i,s_h})$, that includes the market price π_{s_h} corresponding to the state s_h as informed by the market maker, and the information signal $\mathcal{I}_{i,s_h}$. The agent i then uses a *belief update function* to update its beliefs. Finally, agent i selects an action using an *action selection strategy* and receives a reward R_{i,s_h}. The belief update and action selection strategy of the trading agent are discussed next.

4.1.1 Trading agent belief update and utility functions

Recall from Section 4.1 that a belief state of a trading agent is a probability vector that gives a distribution over the set of states S in the prediction market, i.e. $B_{i,h} = (b_{1,h}, ..., b_{|S|,h})$. A trading agent uses its belief update function $b : \Re^{|S|} \times A_i \times O_i \rightarrow \Re^{|S|}$ to update its belief state based on its past action $a_{i,h-1}$, past belief state $B_{i,h-1}$ and the observation o_{i,S_h}. The calculation of the belief update function for each element of the belief state, $b_{s',h}$, $s' \in S$, is described below:

$$
\begin{aligned}
b_{s',h} &= P(s'|a_{i,h-1}, o_i) = \frac{P(s', a_{i,h-1}, o_i,)}{P(a_{i,h-1}, o_i)} \\
&= \frac{P(o_i|s', a_{i,h-1}) \cdot P(s', a_{i,h-1})}{P(o_i|a_{i,h-1})P(a_{i,h-1})}
\end{aligned}
\tag{4.1}
$$

Because $a_{i,h-1}$ is conditionally independent given s' and o_i is conditionally independent given $a_{i,h-1}$, we can rewrite Equation 4.1 as:

$$
\begin{aligned}
& b_{s',h} = \frac{P(o_i|s') \cdot P(s', a_{i,h-1})}{P(o_i)P(a_{i,h-1})} = \\
& \frac{\sum_{\iota \in \mathcal{I}} P(\iota)P(o_i|s', \iota) \sum_{s \in S} P(s)P(s'|s, a_{i,h-1})P(a_{i,h-1})}{P(o_i)P(a_{i,h-1})} \\
= & \frac{\sum_{\iota \in \mathcal{I}} P(\iota)P(o_i|s', \iota) \sum_{s \in S} P(s)P(s'|s, a_{i,h-1})}{P(o_i)} \\
= & \frac{\sum_{\iota \in \mathcal{I}} P(\iota)\Omega(s', \iota, o_i) \sum_{s \in S} T(s, a_{i,h-1}, s')b_{s,h-1}}{P(o_i)}
\end{aligned}
\tag{4.2}
$$

All the terms in the r.h.s. of the Equation 4.2 can be calculated by an agent: $P(\iota)$ is the probability of receiving information signal ι , $\Omega(s', \iota, o_i)$ is the probability of receiving observation o_i in state s' when the information signal is ι, $T(s, a_{i,h-1}, s')$ is the probability that the state s transitions to state s' after agent i takes action $a_{i,h-1}$, $b_{s,h-1}$ is the past belief of agent i about state s, $P(o_i)$ is the probability of receiving observation o_i, which is constant and can be viewed as a normalizing constant.

Incorporating the risk preferences of the trading agents is an important factor in prediction markets. For example, the erroneous result related to the non-correlation between the trader beliefs and market prices in a prediction market in[69] was because the risk preferences of the traders were not accounted for, as noted in[34]. This problem is particularly relevant for risk averse traders because the beliefs(prices) and risk preferences of traders have been reported to be directly correlated[26;62]. Therefore, in our model we assume that the trading agents are risk-averse. The risk preference of an agent i is modeled through a utility function called the constant relative risk aversion (CRRA). We use CRRA utility function to model risk averse agents because it allows to model the effect of different levels of risk aversion and it has been shown to be a better model than alternative families of risk modeling utilities[106]. It has been widely used for modeling risk aversion in various domains including economic domain[49], psychology[66] and in the health domain[8]. The CRRA utility function, $u_i(\phi, R_i)$, for agent i (for legibility we have dropped state k, but the same calculation applies at every state) is given below:

$$
\begin{aligned}
u_i(\phi, R_i) &= \frac{R_i^{1-\theta_i}}{1-\theta_i}, \text{ if } \theta_i \neq 1 \\
&= ln(R_i), \text{ if } \theta_i = 1
\end{aligned} \tag{4.3}
$$

Here, $-1 < \theta_i < 1$ is called the risk preference factor of agent i with $\theta_i > 0$ for risk-averse agents and R_i is the payoff or reward to agent i calculated using the LMSR as was discussed in Section 2.2.2. The reward R_i is calculated after agent i's trade is executed.

4.1.2 Trading agent action selection strategy

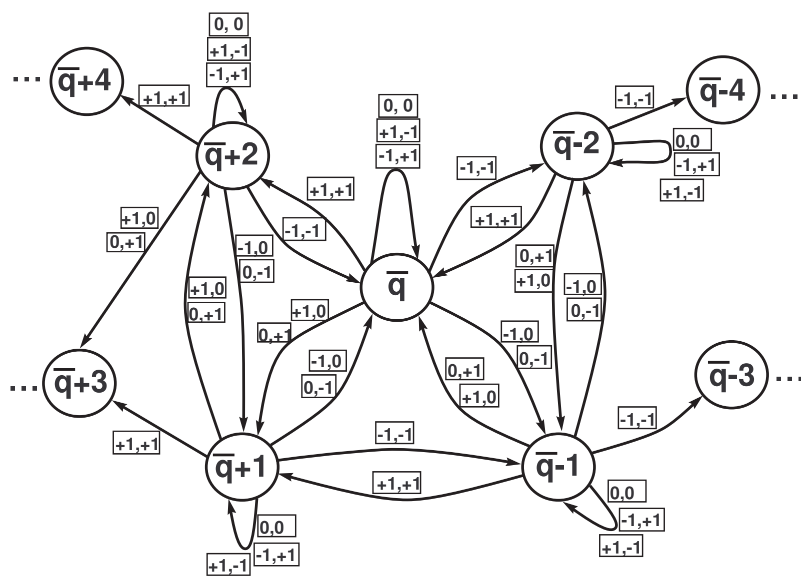


Figure 4.2: Finite state automata of the environment represented by the number of outstanding units of the security, $\overline{q}$, in the prediction market.

The objective of a trading agent in a prediction market is to select an action at each stage so that the expected reward that it receives is maximized. To understand this action selection process, we consider the decision problem facing each trading agent. As an example consider two agents whose available actions during each time step are to buy (=+1) or sell(=-1) only one unit of the security or not do anything (=0) by holding the security. Let the market state be denoted by $\overline{q}$, the number of purchased units of the security. Based on the set of actions available to each agent, the state can transition to one of the following states $\overline{q}+2$ (both agents buy), $\overline{q}+1$ (only one agent buys), $\overline{q}$ (both agents hold, or, one agent buys while the other agent sells, resulting in no transition), $\overline{q}-1$ (only agent sells), and $\overline{q}-2$ (both agents sell). Figure 4.2 shows the finite state automata generated in this way. We can expand this state space further by adding more states and transitions, but the number of states remains finite because the set of states S of the POSGI is finite. Also, since the number of units of a security is finite, the number of securities the trading agent is allowed to buy or sell is bounded and the number of transitions from a state is guaranteed to be bounded. We also do not consider budget restrictions for the trading agents in this work.

4.1.3 Correlated Equilibrium (CE) calculation

In the POSGI, the aggregated price information received by a trading agent from the market maker can be treated as a recommendation signal for selecting the agent's strategy. This situation lends itself to a correlated equilibrium (CE)[2;89], where a trusted external agent privately recommends a strategy to play to each player. A correlated equilibrium is more preferred to the Nash or Bayesian Nash equilibrium because it can lead to improved payoffs, and it can be calculated using a linear program in time polynomial in the number of agents and number of strategies.

Each agent i has a finite set of strategy profiles, Φ_i defined over its action space A_i. The joint strategy space is given by $\Phi = \prod_{i=1}^{|N|} \Phi_i$ and let $\Phi_{-i} = \prod_{j \neq i} \Phi_j$. Let $\phi \in \Phi$ denote a strategy profile and ϕ_i denote player i's component in ϕ. A correlated equilibrium is a distribution p on Φ such that for all agents i and all strategies ϕ_i, ϕ_i' if all agents follow a strategy profile ϕ that recommends player i to choose strategy ϕ_i, agent i has no incentive to play another strategy ϕ_i' instead. This implies that the following expression holds: $\sum_{\phi_{-i}\in\Phi_{-i}} p(\phi)(u_i(\phi) - u_i(\phi_i', \phi_{-i})) \geq 0$, $\forall i \in N$, $\forall \phi_i, \phi_i' \in \Phi_i$ and where $u_i(\phi_i', \phi_{-i})$ is the utility that agent i gets when it changes its strategy to ϕ_i' while all the other agents keep their strategies fixed at ϕ_{-i} and $p(\phi)$ is the probability of realizing a given strategy profile ϕ.

We now prove the existence of a correlated equilibrium in our POSGI-based prediction market.

Theorem 1. *A correlated equilibrium(CE) exists in our POSGI-based prediction market representation at each stage (trading period).*

Proof. At each stage in our prediction market, we can specify the correlated equilibrium by means of linear constraints as given below:

$$\sum_{\phi_{-i}\in\Phi_{-i}} p(\phi)(u_i(\phi) - u_i(\phi_i', \phi_{-i})) \geq 0, \forall i \in N, \forall \phi_i, \phi_i' \in \Phi_i \tag{4.4}$$

$$\sum_{\phi\in\Phi} p(\phi) = 1, \tag{4.5}$$

$$p(\phi) \geq 0 \tag{4.6}$$

Equation 4.4 states that when agent i is recommended to select strategy ϕ_i, it must get no less utility from selecting strategy ϕ_i as it would from selecting any other strategy ϕ_i'. Constraints given in Equations 4.5 and 4.6 guarantee that p is a valid probability distribution. We can rewrite the linear program specification of the correlated equilibrium above by adding an objective function to it.

$$\max \sum_{\phi\in\Phi} p(\phi), \text{ or } \min \ -\sum_{\phi\in\Phi} p(\phi) \text{ s.t.} \tag{4.7}$$

$$\sum_{\phi\in\Phi, \phi_{-i}\in\Phi_{-i}} p(\phi)(u_i(\phi) - u_i(\phi_i', \phi_{-i})) \geq 0, \tag{4.8}$$

$$p(\phi) \geq 0 \tag{4.9}$$

Equation 4.8 is either trivial with a maximum of 0 or unbounded. Next, we show the relationship between Equation 4.5, which defines correlated equilibrium in the form of a constraint program, and the alternate formulation of this problem given in Equation 4.8.

Lemma 1. *Problem given in Equation 4.5 has a solution iff problem in Equation 4.8 is unbounded.*

Proof. If problem in Equation 4.5 has a solution $p(\phi)$, then $p(\phi)$ is also feasible in the problem given in Equation 4.8. However, since for any $a > 1$ $ap(\phi)$ is also feasible in Equation 4.8, but it has a larger value, $p(\phi)$ is not optimal solution in Equation 4.8. Thus, $lim_{a\to\infty} \sum_{\phi\in\Phi}(ap(\phi)) = \infty$ and therefore problem in Equation 4.8 is unbounded.

If the problem in Equation 4.8 is unbounded, its set of solutions is non-empty (by definition). We can transform an arbitrary solution $p'(\phi) \neq 0$ into a solution $p(\phi)$ for problem in Equation 4.5 by normalizing to guarantee that $p(\phi)$ is a valid distribution. □

Lemma 1 shows that there is a correlated equilibrium if and only if problem in Equation 4.8 is unbounded. To prove the unboundedness we consider the dual problem of Equation 4.8 given in Equation 4.11.

$$\max\ 0,\ \text{s.t.} \tag{4.10}$$

$$\sum_{\phi\in\Phi,\phi_{-i}\in\Phi_{-i}} p(\overline{\phi})[(u_i(\phi) - u_i(\phi_i', \phi_{-i})]^T \leq -1 \tag{4.11}$$

$$p(\overline{\phi}) \geq 0 \tag{4.12}$$

where for every $p(\overline{\phi})$ there is $p(\phi)$ such that $p(\phi)[(u_i(\phi) - u_i(\phi_i', \phi_{-i})]p(\overline{\phi}) = 0$.

In[89], the authors showed that the problem given in Equation 4.11 is infeasible. From operations research we know that when the dual problem is infeasible the primal problem is feasible and unbounded. This means that the primal problem from Equation 4.8 is unbounded. We can then conclude that there is at least one correlated equilibrium in every trading period of the prediction market. We note that although the CRRA utility function is concave, the concave structure does not affect the existence of at least one correlated equilibrium because the unboundedness of Equation 4.8 is not affected by the concave structure of u_i. $\square$

To calculate correlated equilibrium(CE) we first characterize the set of all Pareto optimal strategy profiles. A strategy profile ϕ^P is Pareto optimal if there does not exist another strategy profile ϕ' such that $u_i(\phi') \geq u_i(\phi^P)$ $\forall i \in N$ with at least one inequality strict. In other words, a Pareto optimal strategy profile is one such that no trader could be made better off without making someone else worse off. A Pareto optimal strategy profile can be found by maximizing weighted utilities

$$max_\phi \sum_{i=1}^{|N|} \lambda_i u_i(\phi) \text{ for some } \lambda_i \tag{4.13}$$

Setting $\lambda_i = 1$ for all $i \in N$ gives a utilitarian social welfare function. The maximization problem in Equation 4.13 can be solved using the Lagrangian method. We get the following system of $|N|$ equations:

$$\sum_{i=1}^{|N|} \lambda_i \frac{u_i(\phi)}{\phi_i} = 0 \ , \ \forall j = 1, ..., |N| \tag{4.14}$$

that must hold at ϕ^P. Each of these equations is obtained by taking a partial derivative of the respective agent's weighted utility with respect to respective agent's strategy profile, thus solving the maximization problem given in Equation 4.13. By solving the system of equations 4.14 we get the set of Pareto optimal strategy profiles, Φ^P.

We then apply CE calculation algorithm shown in Algorithm 1 on Φ^P. Algorithm 1 is based on the Ellipsoid Against Hope algorithm proposed by[89]. We have used the correction proposed by Stein et al.[90] to solve the numerical precision problems that might arise in the Ellipsoid Against Hope algorithm. The calculation of the matrix values of the U matrix must be done once for each of the N agents. The computation of the utility difference $u_i(\phi) - u_i(\phi_i, \phi_{-i})$ for each agent i can be done in $|\Phi_i^P|^2$ time. Therefore, the time complexity of the CECalc algorithm during each trading period comes to $N \times |\Phi_i^P|^2$.

CECalc(D, Φ^P)
Input: D, Φ^P //D is the duration of the market, Φ^P is the set of Pareto optimal strategies
Output: p //correlated equilibrium
foreach $t \leftarrow 0$ *to* D **do**
  //do this in each trading period
  Let U be the matrix consisting of the values of $(u_i(\phi) - u_i(\phi_i, \phi_{-i}))$, $\forall i \in N, \phi \in \Phi^P, \phi_{-i} \in \Phi_i^P$
  $p'_t \leftarrow$ getDualDistribution(Φ^P, U);
  $p_t \leftarrow$ solve for p_t s.t. $p_t U^T p'_t = 0$;
  return p_t;
end

GetDualDistribution(Φ^P, U)
Input: Φ^P, U
Output: Δ
$l = 0$;
$p'_l \in [0, 1]$;
$\Delta = \{\}$;
while $U^T \cdot p'_l \leq -1$ *is feasible* **do**
  $\Delta = \Delta + p'_l$;
  $p'_{l+1} = p_l + \epsilon^N$; //increase all elements of p' by some small amounts from vector ϵ^N
  $l++$;
end
return Δ;

Algorithm 1: Correlated Equilibrium Algorithm

Proposition 1. *If p is a correlated equilibrium and ϕ is a Pareto optimal strategy profile calculated by p in a prediction market with risk averse agents, then the strategy profile ϕ^P is incentive compatible, that is each agent is best off reporting truthfully.*

Proof. We prove by contradiction. Suppose that ϕ^P is not an incentive compatible strategy, that is, there is some other ϕ' for which

$$u_i(\phi') \geq u_i(\phi^P) \tag{4.15}$$

Equation 4.15 violates two properties of ϕ^P. First of all, since ϕ^P is Pareto optimal, we know that Equation 4.15 is not true, since $u_i(\phi^P) \geq u_i(\phi')$ by the definition of Pareto optimal strategy profile. Secondly, if we rewrite Equation 4.15 as $u_i(\phi^P) - u_i(\phi') \leq 0$ and multiply both sides by $p(\phi^P)$, we get $p(\phi^P)[u_i(\phi^P) - u_i(\phi')] \leq 0$. Since p is a correlated equilibrium this inequality can not hold, otherwise it would violate the definition of the correlated equilibrium. $\square$

4.2 Experimental Results

We conduct several simulations using our POSGI prediction market. The main objective of our simulations is to test whether there is a benefit to the agents to follow the correlated equilibrium strategy. We do this by analyzing the utilities of the agents and the market price. We consider events that are disjoint (non-combinatorial). This allows us to compare our proposed strategy empirically with other existing strategies while using real data collected from the Intrade prediction market, which also considers non-combinatorial events. We report the market price for the security corresponding to the outcome of the event occurring. We assume that risk aversion coefficient of our trading agents, $\theta = 0.6$, since experimental evidence suggests that this value captures humans' risk attitude without being too extreme in either direction[37]. Since currently there is no real data relating to the risk averseness of the human traders and our main goal is to demonstrate the performance of our algorithm, we assume that the trading agents all have the same level of risk aversion.

4.2.1 Intrade Data

For all of our experimental results we use commercially available data from four diverse prediction markets obtained from Intrade[51] company. The details of the Intrade markets that are used for our experiments are given in Table 4.2 and the market prices of these markets in Intrade are shown in Figure 4.3. We note that the date format that is used in Intrade's data and shown in our graphs is day/month/year. We have obtained the general market data comprising of the market price data for each day and also the trade data consisting of the number of shares bought and sold by individual traders and the changes in price after each trade. Intrade prediction market allows for two possible outcomes to

Name	**Description**	**Total number of traders**	**Total number of securities traded**	**Duration** (Trading period)
Presidential Election	"Barack Obama to be elected President in 2008"	3252	1, 244, 892	744 days
Recession	"The US economy to go into recession in 2008"	728	75, 187	600 days
Best Picture	"The Social Network to win Best Picture 2011"	242	18,767	142 days
American Idol	"Lee DeWyze to win American Idol (Season 9)"	105	17,292	93 days

Table 4.2: Intrade markets used for our experiments.

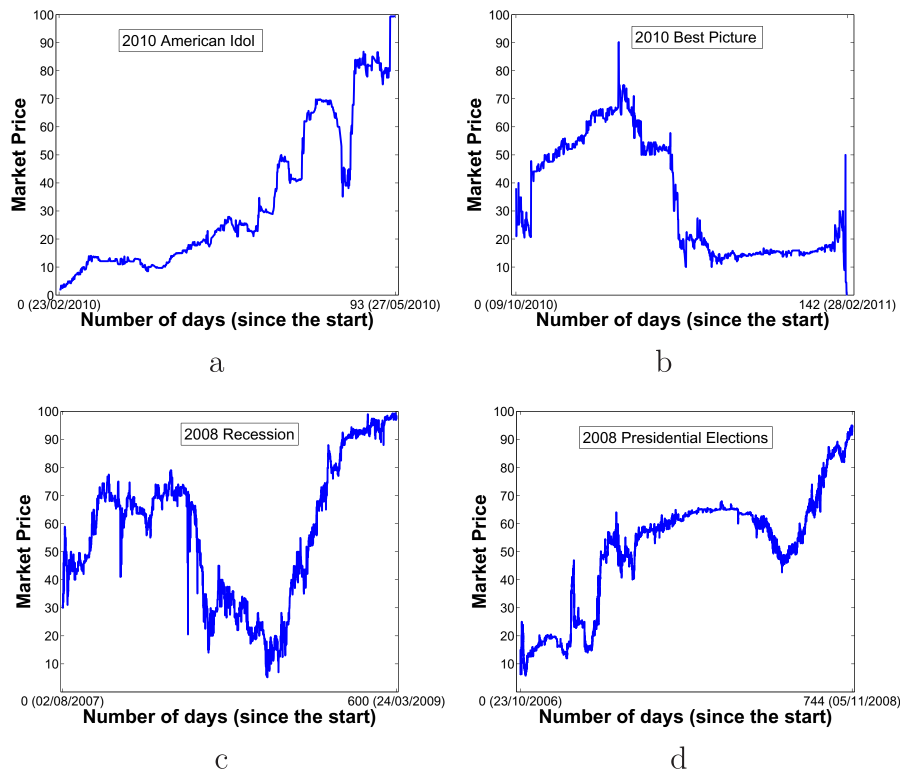


Figure 4.3: Market prices of American Idol(a), Best Picture(b), Recession(c), and Presidential Elections(d) Intrade markets.

each event - yes, the event will happen as described, or no, it will not happen. Intrade is an exchange, so each trader buys or sells securities from another member of Intrade. Therefore, for our experiments we assume that the market maker in our model only calculates the

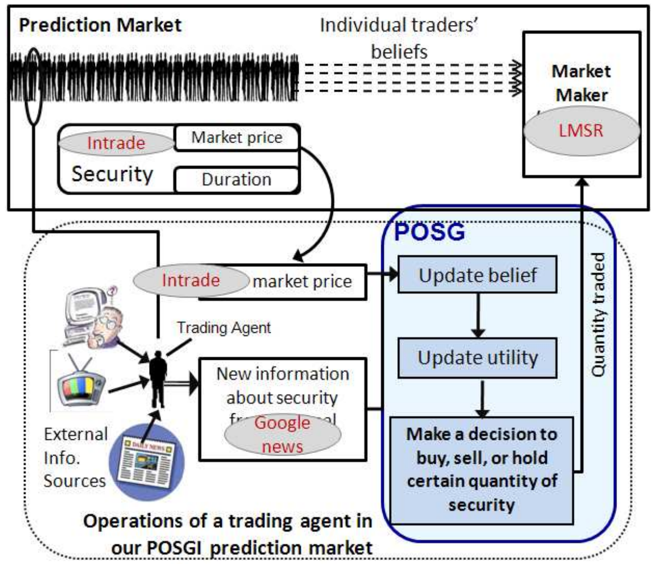


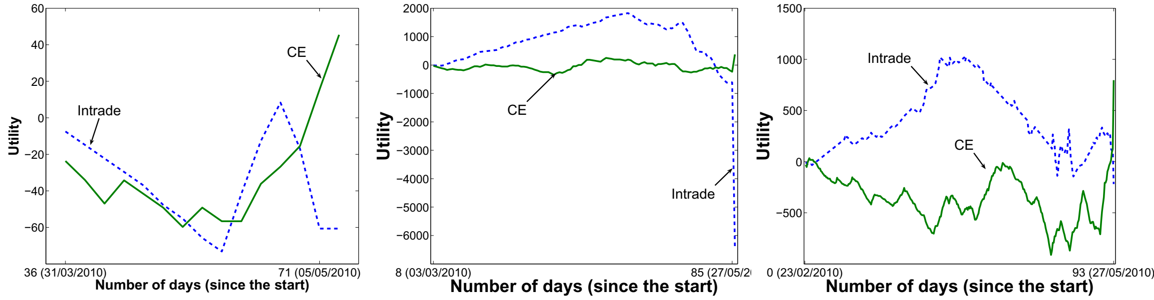
Figure 4.4: Modified diagram of the operations of a trading agent used in our experiments.

market price and the trading agents' rewards, but does not sell or buy securities itself. In Intrade when the market's outcome becomes known, the market settles at either \$0 or \$10. If the market's event happens, the traders holding securities corresponding to the market get paid \$10 for every security that they hold. If the market's event does not happen, the traders with the securities of that market do not get anything. Also, since Intrade allows short selling, i.e. selling securities that you don't yet own, we also allow short selling in our simulated POSGI market.

In all of our simulations we use the same number of trading agents as the number of traders in Intrade markets, and we also sync the start, trade and end times of our trading agents in each simulated market with the start, trade and end times of the real traders from Intrade markets. Figure 4.4 is a modified Figure 3.1 indicating the inputs to and outputs of our POSGI prediction market model used for our simulations. Trading agents receive Intrade's market price according to time steps indicated in the Intrade's data. The market price in Intrade is updated after each trade. We conduct two types of experiments, in the first type of experiments we compare the utility from CE strategy with the utility from the Intrade market, and in the second type of experiments we compare CE strategy to other well-known strategies. For the first type of experiments, in order to be able to compare with the Intrade's data we assume that the trading agents receive Intrade's market price instead of the LMSR market price, but they use CE strategy to determine their action and receive utility based on it. For the second type of experiments, we assume that the market maker uses LMSR to calculate the market price as specified in the POSGI model.

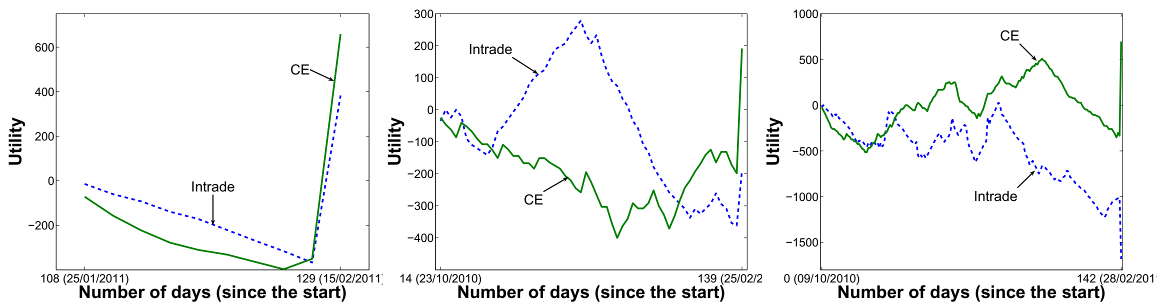
We use Google News[77] to obtain information signals that the trading agents receive. We use each market's description as specified by Intrade to obtain results in Google News, which are then narrowed down by the time period during which each market was active. The results are then ran through SentiStrength[103], which is a tool that estimates the strength of positive and negative sentiment in short texts. We specify positive strength sentiment to be 1, negative strength sentiment to be -1, and neutral strength sentiment to be 0. SentiStrength basically gives a score $(1, -1$ or $0)$ to each result obtained from Google News, which is then used as an information signal sent to a number of the trading agents selected randomly from the total agent population at the same time step (date and time) as the date and time it was published on Google News. Finally, the trading agents in our model update their beliefs either when they receive a new information signal or at the belief update time step interval, which is determined as an average trading time of all traders in the market.

We notice that there were different kind of traders in the Intrade markets, for example some traders were very active and some were not very active, some traders only joined the market for 1 day and some traded in the market throughout its duration. Therefore in order to analyze the performance of different types of trading agents, we have clustered the traders in each of four Intrade markets that we have used according to the duration of their participation in the market and the number of their trades. We use EM(expectation maximization) clustering technique[25] within a popular clustering tool, called Weka 3.6[44]. We didn't specify the number of clusters, instead EM used cross validation to select the number of clusters automatically. We obtained 3 clusters for the American Idol and Best Picture markets, 4 clusters for the Recession market, and 4 clusters for the Presidential Election market.



a b c

Figure 4.5: Average utilities of the traders in Intrade markets and the trading agents in POSGI prediction market for clusters 1 (a), 2 (b), and 3 (c) in the American Idol market.



a b c

Figure 4.6: Average cumulative utilities of traders in Intrade markets and trading agents in POSGI prediction markets for clusters 1 (a), 2 (b), and 3 (c) in the Best Picture market.

For our first set of experiments, we want to compare the performance of the trading agents using CE strategy within POSGI model with the performance of the traders in the real Intrade market. We ran simulations of our POGSI model where all trading agents use CE strategy and we compare their utility to the utility of the actual traders in the Intrade markets using Equation 4.4. We show the results for the average cumulative utility of the traders in Intrade market and the average cumulative utility of the trading agents in the POSGI market for each cluster in each market. The last point in the utility graphs corresponds to the final utility that the trading agents receive after the market clears, i.e. trading agents get paid \$10 for each security they possess at the end of the market.

Figures 4.5 and 4.6 (a-c) show the average utility of the trading agents for cluster $1, 2,$ and 3 correspondingly in the American Idol and the Best Picture markets. While Figures

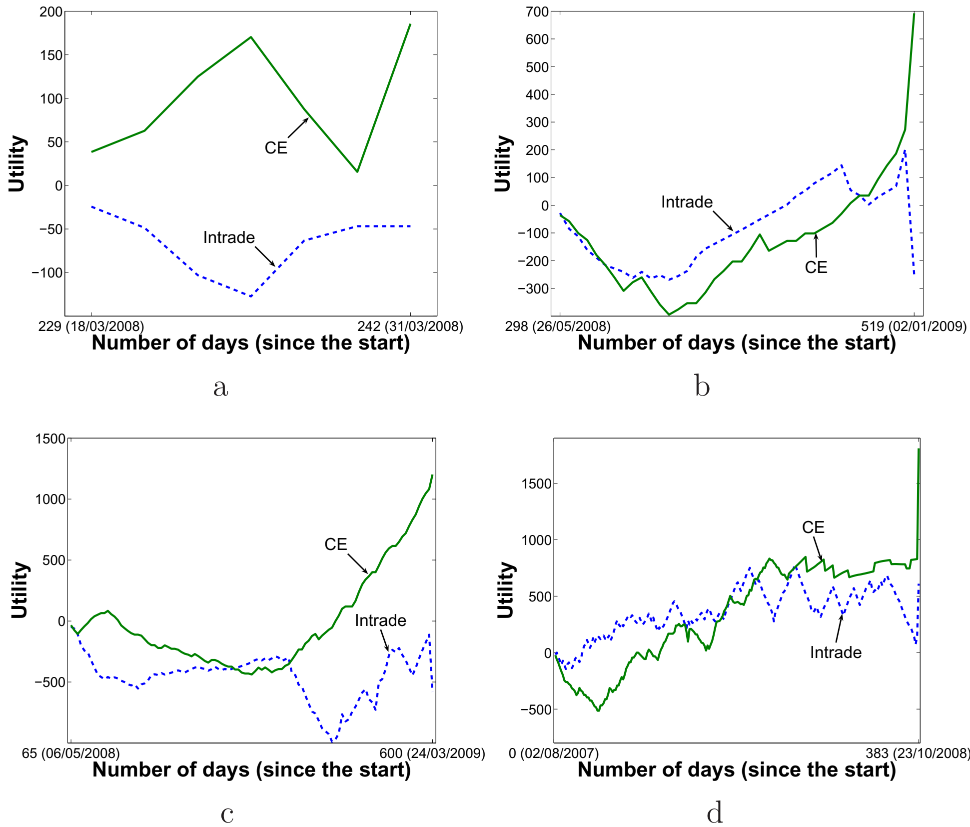


Figure 4.7: Average utilities of traders in Intrade markets and trading agents in POSGI prediction markets for clusters 1 (a), 2 (b), 3 (c), and 4 (d) in the Recession market.

4.7 and 4.8 (a-d) show the average utility of the trading agents for cluster $1, 2, 3$, and 4 correspondingly in the Recession and the Presidential Election markets. We note that different clusters have different durations, for example in the Best Picture market, cluster 1 shown in Figure 4.6(a) lasts for only 21 days, whereas cluster 3 shown in Figure 4.6(c) lasts for the entire duration of the market. From this set of experiments we observe that due to the look-ahead capability of the CE strategy the trading agents are able to get more utility than the human traders in the Intrade markets. A negative trading agents' utility throughout most of the duration in some markets is because the agents buy securities at the beginning of the market's duration and in the markets(or clusters) with a shorter duration, such as the American Idol market, they do not have enough time to play the market to increase their utility until almost the end of the market. However, in the markets with

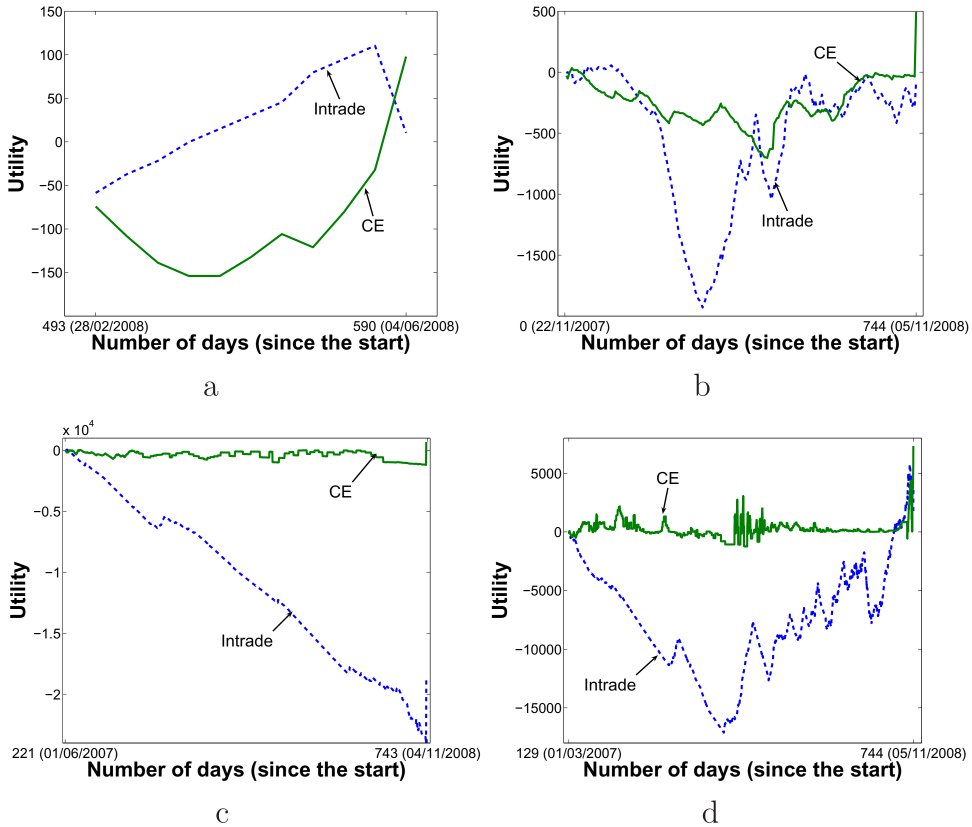


Figure 4.8: Average utilities of traders in Intrade markets and trading agents in POSGI prediction markets for clusters 1 (a), 2 (b), 3 (c), and 4 (d) in the Presidential Election market.

longer duration the trading agents are able to increase their utility throughout the market's duration. In summary of this set of experiments, we observe that using trading agents with CE strategy to trade on behalf of humans may be beneficial since it leads to higher utility and it can avoid inefficient human trading decisions that might result in a very large loss, such as the one observed in the Presidential Election market for cluster 3 shown in Figure 9 (c).

For our next set of experiments, we compare the trading agents' and market's behavior under various strategies employed by the trading agents in Intrade's markets given in Table 4.2. In this set of experiments in each run of each market all agents use the same trading strategy. We then compare separate runs with different strategies. We use the following five well-known strategies[67] for comparison with our proposed CE strategy.

a. ZI (Zero Intelligence) - each agent submits randomly calculated quantity to buy or sell.

b. ZIP (Zero Intelligence Plus) - each agent selects a quantity to buy or sell that satisfies a particular level of profit by adopting its profit margin based on past prices.

c. CP (by Preist and Tol) - each agent adjusts its quantity to buy or sell based on past prices and tries to choose that quantity so that it is competitive among other agents.

d. GD (by Gjerstad and Dickhaut) - each agent maintains a history of past transactions and chooses the quantity to buy or sell that maximizes its expected utility.

e. DP (Dynamic Programming solution for POSG game) - each agent uses dynamic programming solution to find the best quantity to buy or sell that maximizes its expected utility given past prices, past utility, past belief and the information signal[45].

Market	p **value of the t-test**
Presidential Election	0.07
Recession	0.11
Best Picture	0.08
American Idol	0.12

Table 4.3: Correlation test conducted using two tailed type 3 t-test showing the correlation of the market prices in the Intrade markets and the market prices produced by LMSR in POSGI prediction markets.

In order to compare the effect of different trading strategies on the market prices, in the remaining set of experiments we have LMSR market maker calculate the market price and send the LMSR market price back to the trading agents instead of the Intrade's market price. We first compare the market prices calculated by LMSR market maker and the real Intrade's market prices. The results of the t-test shown in Table 4.3 indicate a strong correlation between the two market prices for all four markets.

Figures 4.9 (a) column show the market prices calculated by the LMSR market maker for four Intrade markets. We observe that agents using the CE strategy are able to trade at prices that are closer to the final outcome of the event, indicating that agents using

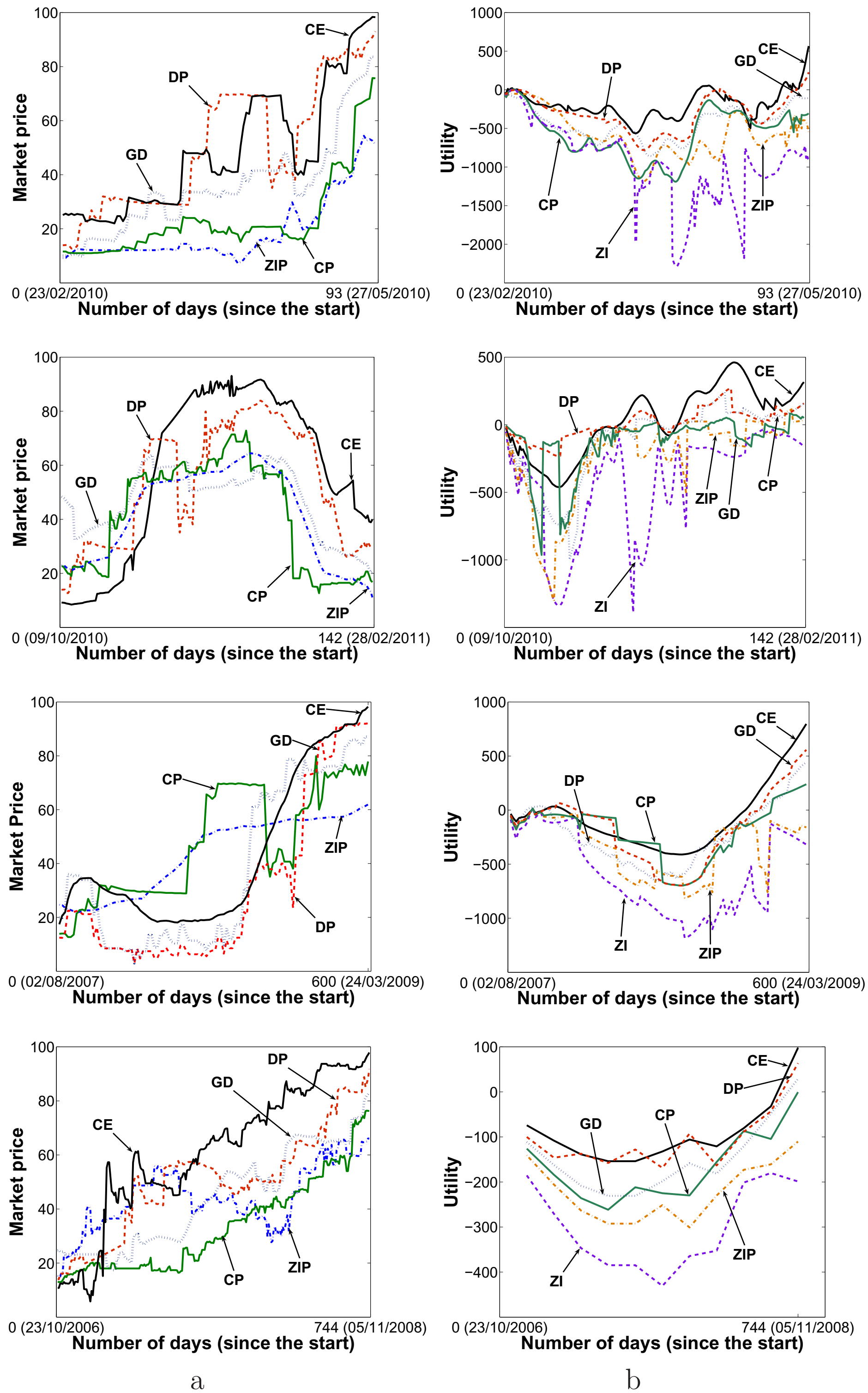


Figure 4.9: The market prices(a) and the average utilities(b) of the agents under different trading strategies for American Idol market, Best Picture market, Recession market, and Presidential Election market (top to bottom).

the CE strategy are able to respond to other agents' strategies and predict the aggregated price of the security more efficiently. This efficiency is further supported by the graph in Figures 4.9(b) column that show the utilities of the agents while using different strategies. The agent population was uniformly divided for different strategies. We see that the agents using the CE strategy are able to obtain 39% more utility on average than the agents following the next best performing strategy (DP) and 137% more utility on average than the agents following the worst performing strategy (ZI) in all markets. Overall, we can say that using the POSGI model allows the agent to avoid myopically predicting prices and use the correlated equilibrium to calculate prices more accurately and obtain higher utilities.

In summary, the POSGI model and the CE strategy proposed in this chapter of the thesis result in better price tracking and higher utilities because they provide each agent with a strategic behavior while taking into account the observations of the prediction market and the new information of the events. The work presented in this part of the thesis was published in [56] and [54] and the overview of our findings is shown in Table 4.4.

Contr.	Research Question	Research Findings
2	How do different trading agent behaviors affect the behavior of prediction markets?	- We observed that a higher utility of the trading agents leads to a more accurate price prediction.
2	What trading strategies perform the best in prediction markets?	- Our proposed strategy based on Correlated Equilibrium was shown to perform the best.
2	How can prediction markets incentivize trading agents to participate in order to achieve a higher prediction accuracy?	- By rewarding (paying) trading agents for useful trades and penalizing them for misleading trades.
2	How does the trader's behavior using a formal game-theoretic model compare to the trader's behavior in real prediction markets?	- Our experiments and correlated tests show that there is a strong correlation between them.

Table 4.4: Research findings for research questions in contribution 2 of this thesis.

Chapter 5

A Multi-Agent Prediction Market Based on Boolean Network Evolution for Analyzing Market Dynamics

In this chapter of the thesis, we propose a form of a dynamical system, called a Boolean Network (BN), that uses simple Boolean rules to model the operation of a prediction market. We then use this BN-based model to study the overall dynamics of the prediction market and how various parameters effect its behavior. In a BN, each node is represented by a binary state while the network edges represent rules that update the state of the node that the edges are incident on. Although inherently simple, BNs can be used to analyze essential aspects of complex networks such as values of parameters that effect a specific behavior and the time required to reach that behavior. It also makes sense to use Boolean networks in the context of a prediction market because there is a direct correspondence between Boolean values output by the Boolean network's rules and the binary outcomes of events predicted by a prediction market.

The main contributions of this part of thesis are to develop simple Boolean rules for updating the beliefs for each of the market's participants and for aggregating the participants' belief information into a single market price. We show that despite the simplification of the traders' beliefs in the prediction market into Boolean states, the aggregated market price calculated using our BN model is strongly correlated with the price calculated by a commonly used aggregation strategy in existing prediction markets called the Logarithmic Market Scoring Rule (LMSR). Our experimental results show that our BN model also eliminates the problem of frequently fluctuating prices that are known to be a drawback of the LMSR. We also use our BN model to analyze the dynamics of the prediction market with respect to different market parameters and determine the conditions under which the market price converges. Finally, we model the untruthful belief revelation by the market participants, a commonly encountered problem in prediction markets, using the presence of noise in the Boolean rules of our prediction market and obtain similar results as the conventional (non-Boolean) prediction markets. The research questions that we attempt to answer in this contribution are reproduced below in Table 5.1.

Contr.	Research Question	Research Approach
3	How does a prediction market evolve and what are its dynamics under different market and trader conditions?	Theoretical Examination Empirical Analysis
3	How can we make a prediction market unaffected by "noise"?	Theoretical Examination Empirical Analysis

Table 5.1: Research questions that are addressed in contribution 3 of this thesis.

5.1 BN-based Prediction Market

As our previous multi-agent prediction markets, our BN-based prediction market consists of three major entities: trading agents, a market maker agent, and information sources that are external to the market but provide information to the market's agents. Because of the

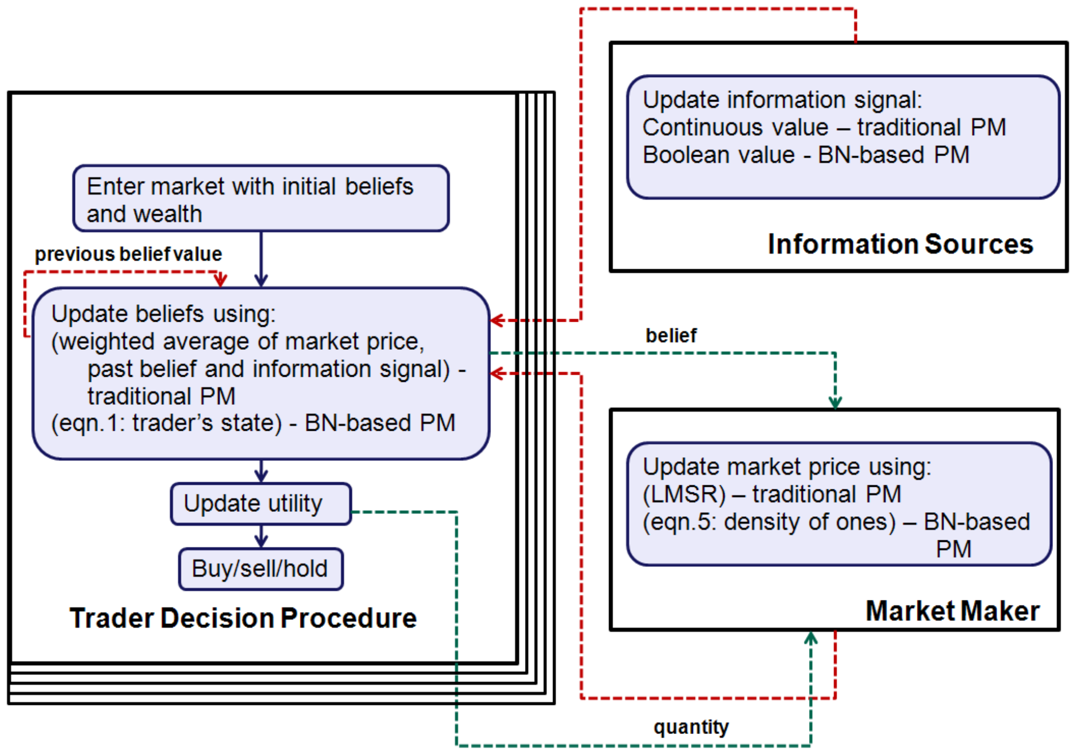


Figure 5.1: The sequence of operations done by the trading agents and the market maker agent in one trading period in a prediction market.

binary nature of the event outcomes, it makes sense to use Boolean functions to represent the beliefs of the traders in prediction markets[21]. The basic operations of our BN-based prediction market are based on the traditional prediction market's operations, however the trading agents' beliefs are updated using a Boolean function and a novel technique using the Boolean beliefs of the trading agents is used to calculate the market price. Figure 5.1 shows the operation of both the conventional and the BN-based prediction market proposed in this part of the thesis *. To do this, each trading agent maintains a belief about the outcome of the security corresponding to the event and updates this belief at certain intervals using the aggregated market price, past belief values and external information. In our BN-based prediction market each trading agent uses a variable called a *state* to represent this belief. Each state can take one of two values: 1 or ON, meaning that the trading agent believes

*For the simplicity of our discussion and without the loss of generality, we assume there is one event in our prediction market with two possible outcomes - event happens/does not happen.

that the event will happen, or 0 or OFF, meaning that the trading agent believes the event will not happen. Following the belief update rule in a conventional prediction market, trading agents update the value of their state at each trading period t based on the current aggregated market price, their past state, and the information signal they receive about the event. The state update procedure is represented as a Boolean function which is described in the next section. After the trading agents update their state, they calculate their expected utility using their past state and the current market price and use this utility to determine the optimal quantity of each security to buy or sell. The optimal quantity to buy or sell is given by the quantity that maximizes the expected utility of the trading agent. The trading agents send the quantity of securities they want to buy or sell and their current belief/state to the market maker agent. The market maker agent updates the market price after aggregating the beliefs received from the trading agents. The market maker agent also calculates the cost of each trading agent's transaction and sends it back to each trading agent.

In the next section we describe the Boolean function formulation of the operations by the trading agents and the market maker agent in a prediction market. A summary comparison between the operations of our BN-based prediction market and a conventional (LMSR-based) prediction market is given in Table 5.2.

5.1.1 Trading Agents' Boolean Belief Update

The state of a trading agent $n \in N$ is determined by the three variables defined below:

a. $p_r(t)$ - the aggregated market price at trading period t. In our BN-based prediction market we call the current aggregated market price the *density of ones* which is the fraction of trading agents that are in state 1 at a given trading period t.

b. $r_n(t)$ - state of the n-th trading agent at trading period t representing its belief about the outcome of the event.

c. $\overline{w^n} = (w_1^n, w_2^n, w_3^n)$ - a vector of weights representing the trust that the n-th trading agent holds for the accuracy of the posted market price, its own past belief, and

Operation	Conventional PM	BN-based PM
Belief update	1. Trading agents calculate their beliefs as a weighted average of the market price, their past belief and the information signal[109] with all the parameters $\in [0,1]$.	1. Trading agents' beliefs are represented through their Boolean states which are updated as a threshold function of the weighted average of the market price $\in [0,1]$, their past Boolean state $\in \{0,1\}$ and the Bernoulli information signal $\in \{0,1\}$.
	2. Trading agents submit their beliefs as a discrete value $\in [0,1]$.	2. Trading agents send their belief (i.e. their state) as a Boolean value $\in \{0,1\}$.
Aggregation rule	3. The market maker uses some rule such as LMSR to aggregate the beliefs of the traders and set the market price[13].	3. The market maker uses the fraction of traders that are ON to calculate the market price.
External information service	4. Most prediction markets use a continuous probability distribution to model the external information signal.	4. Following[53], we use a Boolean value for the signal.

Table 5.2: Differences between conventional LMSR-based prediction market and our BN-based prediction market.

the new information signal it obtains, respectively, following[38]. These trusts are represented as weights $w_i^n \in [0,1]$, such that $\sum_{i=1}^{3} w_i^n = 1$, for $i = 1,2,3$.

Let $B_n(t)$ be the information signal received by the n-th trading agent at trading period t. For simplicity and for the purpose of illustration of this method, we assume that $B_n(t)$ is the value of a Bernoulli random variable with probability q_n of obtaining a 1, that is positive information, and probability $1 - q_n$ of obtaining a 0, that is negative information. The rule that generates the new state of the n-th trading agent can be written as follows and is shown as a diagram in Figure 5.2:

$$r_n(t+1) = \begin{cases} 1 & \text{, if } w_1^n \cdot p_r(t) + w_2^n \cdot r_n(t) + w_3^n \cdot B_n(t) > z, \\ & \sum_{i=1}^{3} w_i^n = 1,\ w_i^n \in [0,1], \text{ for } i = 1,2,3; \\ 0 & \text{, otherwise.} \end{cases} \tag{5.1}$$

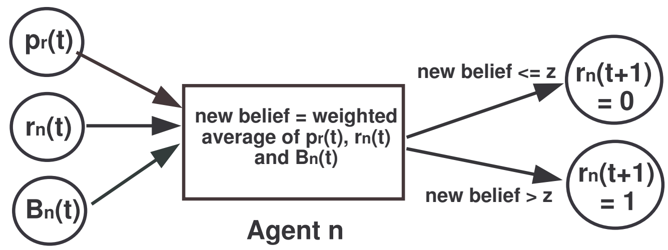


Figure 5.2: The Boolean belief update used by agent n at trading period t.

Here $z \in [0,1]$ represents a threshold parameter used to convert the quantity $w_1^n \cdot p_r(t) + w_2^n \cdot r_n(t) + w_3^n \cdot B_n(t)$ into a Boolean value. The rule basically says that the trading agent r_n is turned ON at trading period $t+1$ if the weighted sum of the market price, its own past state, and the external information signal is greater than some threshold value z at trading period t. Thus the trading agent rule is a linear threshold function. The value of z indicates the boundary between what is considered negative or positive overall impact of the aggregated information on each trading agent's belief. For simplicity, we will assume that z is fixed for all trading agents. Although in real prediction markets different agents may have different ways of evaluating information and reflecting on their past experiences, for simplicity we assume that the trust weights $\overline{w^n}$ and the Bernoulli distribution B are the same for all trading agents. Also, in real prediction markets different trading agents may have different thresholds or predispositions for believing that an event will take place; therefore future work will allow for generalizations with varying thresholds. In Section 5.2.1 we show how the weights, w_1^n, w_2^n, w_3^n, can be learned using a neural network.

5.1.2 Mean-field Analysis for Calculating the Aggregated Market Price by Market Maker agents

The fraction of trading agents in state 1 at trading period t give the aggregated belief of trading agents that believe the event will happen at trading period t. In our model this value is represented through the density of ones which is calculated by the market maker agent. The market maker agent uses a mean-field approach specific to statistical physics to generate a recursive mathematical model for the density of ones. The mean-field approach assumes a sufficiently large number of nodes so that potential local correlations can be ignored. This makes the computations more manageable.

Let $p_r(t)$ be the probability that a (generic) trading agent is ON at trading period t, and $1 - p_r(t)$ the probability that the trading agent is OFF at trading period t. We find $p_r(t+1)$ in terms of $p_r(t)$, using a probabilistic approach typical for derivations of mean-field formulae, based on the law of total probability and the assumption of independence of inputs of the rules governing the dynamics of the prediction market. Since the trust weights $\overline{w^n}$ and Bernoulli distribution $B_n(t)$ is assumed to be the same for all agents, in the derivation below we will drop the trading agent index n.

Observe that by the rule of total probability, $p_r(t+1) = P(r(t+1) = 1) = P(r(t+1) = 1|r(t) = 0)(1 - p_r(t)) + P(r(t+1) = 1|r(t) = 1)p_r(t)$, where $P(A)$ is used to denote the probability of an event A. We note that

$$P(r(t+1) = 1|r(t) = 0) = P\left(w_1 p_r(t) + w_3 B > z\right) = \tag{5.2}$$

$= P\left(B > \frac{z - w_1 p_r(t)}{w_3}\right)$ and similarly

$$P(r(t+1) = 1|r(t) = 1) = P\left(w_1 p_r(t) + w_2 + w_3 B > z\right) = \tag{5.3}$$

$= P\left(B > \frac{z - w_1 p_r(t) - w_2}{w_3}\right)$.

Putting equations (5.2) and (5.3), we get:

$$p_r(t+1) = P\left(B > \frac{z - w_1 p_r(t)}{w_3}\right)(1 - p_r(t)) + \tag{5.4}$$

$+P\left(B > \frac{z - w_1 p_r(t) - w_2}{w_3}\right) p_r(t)$.

To simplify the notation, denote $F_B(b) = P(B > b)$, the complementary cumulative distribution function associated to the random variable B. Then the formula for $p_r(t+1)$ becomes

$$p_r(t+1) = F_B\left(\frac{z - w_1 p_r(t)}{w_3}\right)(1 - p_r(t)) + \tag{5.5}$$

$+F_B\left(\frac{z-w_1p_r(t)-w_2}{w_3}\right)p_r(t)$.

Observe that this formula can be used with both discrete and continuous distributions for the external information. However, in the numerical investigations we will focus on the Bernoulli random variable. For the Bernoulli case, we can actually compute the values of F_B according to the relative positions of $\frac{z-w_1p_r(t)-w_2}{w_3} < \frac{z-w_1p_r(t)}{w_3}$ with respect to the two possible values of B, namely 0 and 1. Recall that q is the probability that B is 1. By a straightforward computation we obtain:

$$p_r(t+1) = \begin{cases} 1, \text{ if } p_r(t) > \frac{z}{w_1} \\ q(1-p_r(t)) + p_r(t), \text{ if} \\ \quad \max\{\frac{z-w_2}{w_1}, \frac{z-w_3}{w_1}\} < p_r(t) \leq \frac{z}{w_1} \\ \begin{cases} p_r(t), \text{ if } \frac{z-w_2}{w_1} < p_r(t) \leq \frac{z-w_3}{w_1} \\ q, \text{ if } \frac{z-w_3}{w_1} < p_r(t) \leq \frac{z-w_2}{w_1} \end{cases} \\ qp_r(t), \text{ if } \frac{z-w_2-w_3}{w_1} < p_r(t) \leq \min\{\frac{z-w_2}{w_1}, \frac{z-w_3}{w_1}\} \\ 0 \qquad, \text{ if } p_r(t) \leq \frac{z-w_2-w_3}{w_1}. \end{cases} \tag{5.6}$$

The mathematical model for the density of ones not only represents the aggregated market price but can also be used to analyze the dynamics of the prediction market. Observe that the function (5.6) represents a *map* (that is a function whose domain and codomain are the same) on $[0,1]$ whose *fixed points* can be computed. Let us denote it by $f(p)$. A point $p \in S$ is a fixed point of the map f if $f(p) = p$. It is known from chaos theory that the fixed points of a map drive the dynamics of the map. More precisely, if say p is a fixed point of f, then if $|f'(p)| < 1$, the fixed point p attracts all other points close enough to p. More precisely, if x is a point close to p, then $f^n(x) \to p$ as $n \to \infty$, where $f^n(x)$ is the n-th iterate of f at the point x. The set $\{x, f(x), f^2(x), ..., f^n(x), ...\}$ is called the orbit of x. On the other hand, if $|f'(p)| > 1$, the fixed point p repels all the orbits starting at points x in a neighborhood of p.

We find the fixed points for the map given in (5.6) in our BN-based prediction market by setting $p_r(t+1) = p_r(t)$. The analysis of the stability of the fixed points of the map (5.6) reveals that the fixed points 0 and 1 are always stable. On the other hand, if $w_2 < w_3$ then there is a third stable point q. The orbits will be attracted to one of these three fixed points, depending on the parameters. If $w_2 > w_3$, we may also end up with the case where all points in $[0, 1]$ are fixed points, which means that all states are frozen from the very beginning, so the system is unstable. This can happen if $\frac{z-w_2}{w_1} < 0$ and $\frac{z-w_3}{w_1} > 1$ which means $w_1 + w_3 < z < w_2$. Higher order iterations of the map (5.6) do not reveal more complexity. Thus, in case the external information is modeled by a Bernoulli random variable, the behavior of the model is non-complex and can be easily predicted.

5.2 Experimental Results

5.2.1 Learning the trust values by trading agents

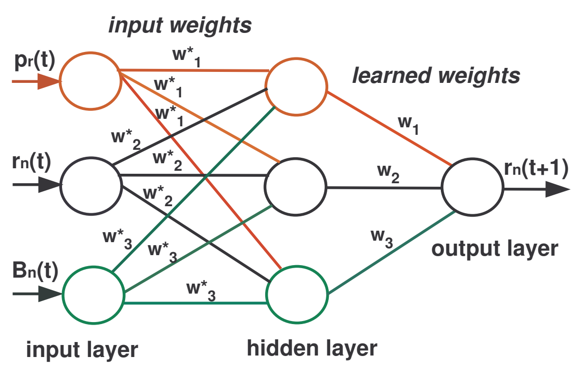


Figure 5.3: One hidden layer neural network used for learn trust weights.

To find the correct combination of weight parameters, w_1^n, w_2^n, w_3^n used by the n-th trading agent's belief update rule given in Equation 5.1, we use a neural network representation. We construct a neural network with one hidden layer, where the market price at trading period t, $p_r(t)$, the state of the trading agent at trading period t, $r_n(t)$, and the Bernoulli variable representing the information parameter, $B_n(t)$, are the inputs to the network. The new state of the trading agent at trading period $t+1$, $r_n(t+1)$, is the output of the neural network. The representation of the neural network used is shown in Figure 5.3. The initial input weights are set randomly, while the learned (output) weights are learned for different values of the parameters in our BN model, namely z - the threshold parameter, and q - the probability that the Bernoulli random variable is 1. We use the backpropagation technique to learn the weights in our neural network[97]. The training set used for the neural network was obtained by simulating the prediction market for over 200 different combinations of values of z and q parameters in our BN. For the data generated for the training set $p_r(t)$ was calculated as the fraction of the trading agent nodes that are equal to 1 at time t, $r_n(t)$ was set to 1 if the belief that maximizes the expected utility of the trading agent[34] was above z and 0 otherwise, and $B_n(t)$ was set to 0 or 1 based on the value of q. A set of learned weights was generated for each combination of z and q values. The learned values are used in the numerical investigations given below.

5.2.2 Patterns and validation of the mean-field based price aggregation mechanism

Having too few trading agents may lead to discrepancies between the mathematical mean-field model and the actual simulation of prediction market due to the fact that for a mean-field approximation the prediction market has to be large enough to ignore local correlations. In our experiments we found that a prediction market with 100 trading agents is sufficiently large for a good match in the fraction of trading agents that are ON between the mathematical model and the actual network. Therefore, in all of our experimental results except those presented in Section 5.2.5 we use 100 trading agents. Similar results were obtained with a larger number of trading agents. We start our experimental analysis by presenting *pattern formation plots* generated with a Boolean network governed by the rules presented in Section 5.1.1. More precisely, pattern formation plots are obtained by arranging the nodes, representing the trading agents, in a one-dimensional array and numbering them from left to right. Then we choose an initial state of the prediction market and iterate it a number of trading periods with time evolving downwards. We plot a black dot when the state of the trading agent is 1 and a neutral dot when it is 0. Figure 5.4 shows the

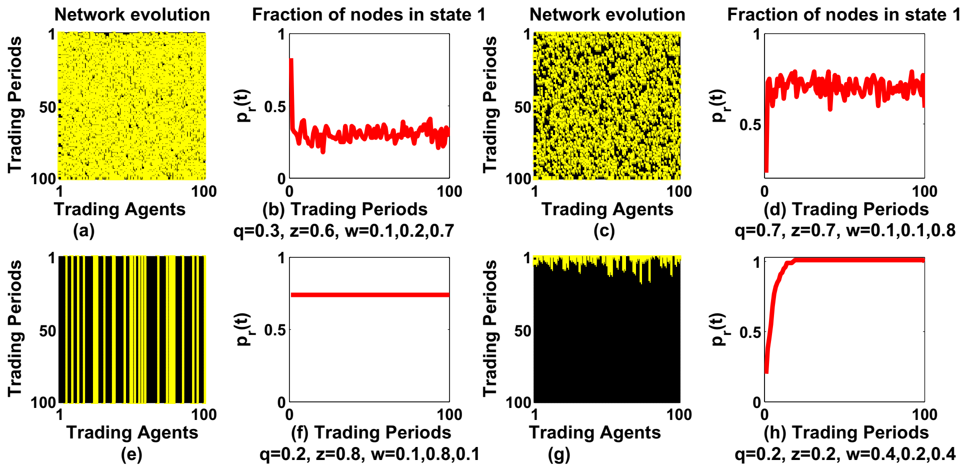
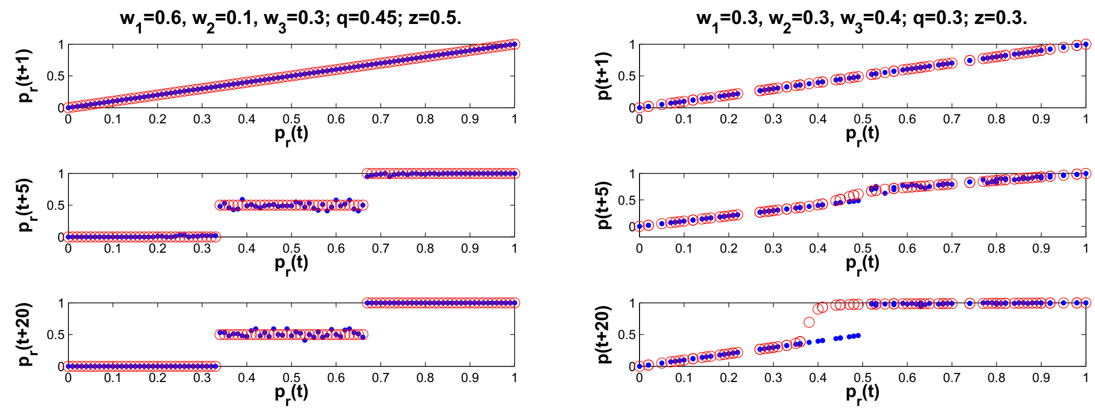


Figure 5.4: (a),(c),(e),(g): Pattern Formation plots for a prediction market starting with a random initial condition and the parameters specified in the associated right plots. (b),(d),(f),(h): The corresponding aggregated market price. The parameters are set as specified in the graphs.

pattern formation plots and the corresponding aggregated market price using BN at each trading period of the prediction market's evolution. This is done for four distinct parameter combinations. We can see in Figure 5.4(a), that for a low value of $q = 0.3$ and a medium value of $z = 0.6$ (which means that the most weight is given to the information value), the aggregated market price oscillates in a narrow range of values around 0.3. The corresponding pattern formation plot in Figure 5.4(b) looks random but with more nodes in state 0 (more neutral dots). Figures 5.4(c) and (d) show similar result for $q = 0.7$ and $z = 0.7$. Here the aggregated market price does not reach stability, but it oscillates within a narrow range of values around 0.7 and therefore its corresponding pattern has more nodes in state 1 (more black dots). The overall higher values for the aggregated market price are due to the fact that the probability of information signal being 1 is high ($q = 0.7$) and the weight corresponding to the information signal is also high ($w_3 = 0.8$). In Figures 5.4(e) and (f) the parameters are $q = 0.2$ and $z = 0.8$. We can see that the aggregated market price is stable around 0.7 and thus the pattern is stationary with neutral and black vertical stripes showing that trading agents are either in state 0 or in state 1 throughout the prediction market's duration. Finally, for the parameters $q = 0.2, z = 0.2$ in Figure 5.4(g) and (h) it takes less than 25 trading periods for the aggregated market price to reach stability. This



a b

Figure 5.5: The system (blue dots) versus the mathematical model (red circles) for the 1st, 5th, and 20th trading periods. Note the apparent match between them.

can be seen more clearly from the pattern formation plot where the top of the plot shows clear randomness, while the rest of the plot is black meaning all of the trading agents are in state 1. The aggregated market price is able to converge here mostly because of the low value of the threshold parameter z. Thus from these results we can see that the aggregated market price can be used as a predictor for future market dynamics. It can also estimate the trading period needed to reach a certain type of long-term behavior, e.g. convergence.

We now check that the mean-field model for the aggregated market price, $p_r(t)$, derived in Equation (5.5) is a good approximation for the fraction of nodes in state 1 obtained by evolving the actual BN. We do this by graphing on the same plot both $p_r(t)$ and the actual fraction of trading agents in state 1 for the 1st, 5th, and the 20th trading periods as shown in Figure 5.5. On the x-axis we plot the initial conditions for the fraction of trading agents in state 1 ($\{0, \frac{1}{N}, \frac{2}{N}, ..., , \frac{(N-1)}{N}\}$), representing how many traders are initially in state 1, i.e. believe that the event will happen. We first apply the mathematical model to each of these initial conditions, iterate them, and plot the results with a red straight line. We then apply the prediction market evolution for a network state corresponding to each initial fraction of the trading agents in state 1, evolve the prediction market, and plot it with a blue '+'. For each given initial fraction of ones we randomly select trading agents that are in state 1. Figure 5.5 shows the comparison results for two different combinations of the parameters

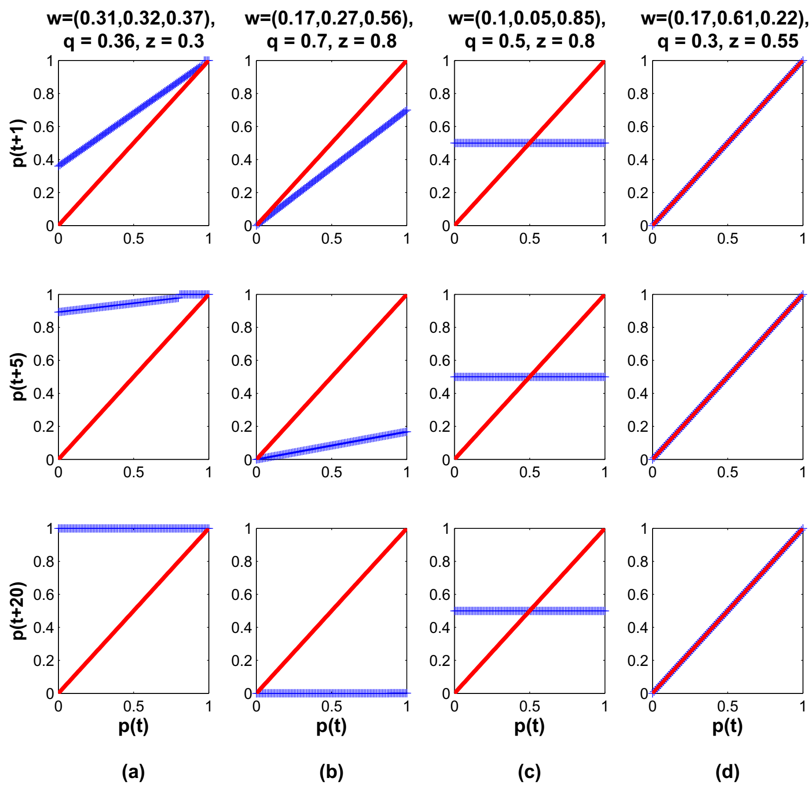


Figure 5.6: Graphical illustration of the iterations of the aggregated market price (density of ones map given in Equation 5.6) (blue '+') versus the main diagonal (red line). The intersection between them yields the fixed points of the map.

w, q, and z. We performed exhaustive simulations for the possible ranges of q, z, and their corresponding weights learned via neural network, and obtained similar results to those in Figure 5.5. We can see from Figure 5.5 that the first iteration matches perfectly. Then, as the prediction market and the mathematical model evolve during their transient phase, the match becomes a little less perfect due to the actual correlations that are building up in a prediction market. These correlations are ignored in the mean-field approach. Despite the assumption of no correlations, in the long run the mathematical model for $p_r(t)$ is a very good approximation of the evolution of the aggregated market price for the actual network.

We also illustrate the behavior of our mean-field based model for the aggregated market price using BN by generating multiple iterations of the mathematical model for $p_r(t+1)$ (blue line marked with '+') for various values of q and z in Figure 5.6. We also plot the line representing $p_r(t+1) = p_r(t)$ (red straight line). Note that the intersection of each iteration with the first diagonal generates the fixed points. As we discussed in Section 5.1.2

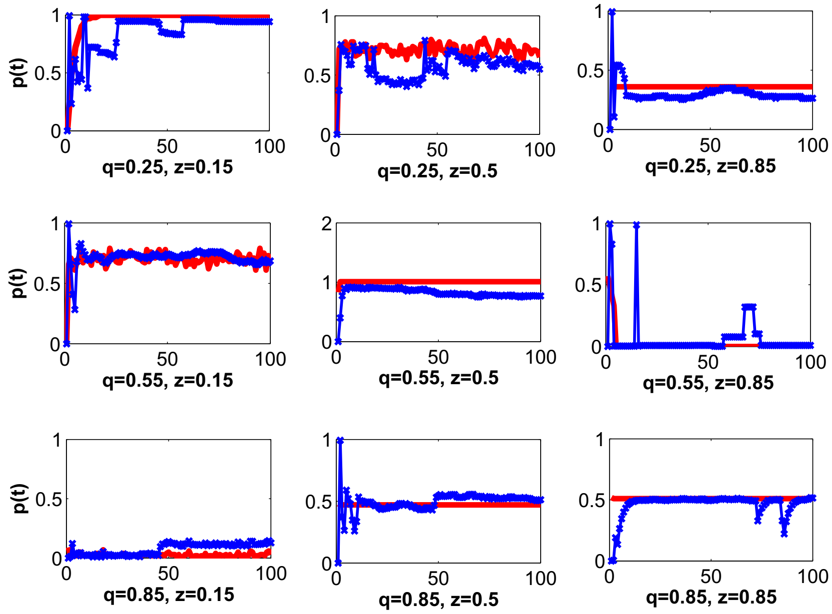


Figure 5.7: Comparison of the market prices set by LMSR (marked by 'x' - blue) and the aggregated market price using BN (marked by '⋄' - red) by trading periods. Note the similarities between the two models, as well as the robustness of the BN model as opposed to the increased variation of the LMSR model.

our system has fixed points (when $p_r(t+1) = p_r(t)$) at 0, 1 and q. We find that our prediction market mainly conforms to one of four behaviors shown in Figure 5.6. Figure 5.6(a) shows the case when the system converges to 1, while Figure 5.6(b) illustrates the case when the prediction market converges to 0, and Figure 5.6(c) shows the existence of the fixed point at q. Figure 5.6(d) shows the last case when $p_r(t+1) = p_r(t)$, so the two lines overlap, meaning chaos.

5.2.3 Comparison to Conventional Prediction Markets

In this section we compare the aggregated market price obtained using the BN-based prediction market model to the aggregated market price obtained with a Logarithmic Market Scoring Rule (LMSR) aggregation mechanism[13] while using the same underlying parameters. To illustrate the comparison we graph both market prices on the same plot for different values of q and z. The x-axis represents the number of trading periods, while the y-axis is the market price. We can see from Figure 5.7 that in the long run, both the LMSR and BN models yield approximately the same results. It is also revealed that the aggregated market price does not fluctuate as much as the LMSR price, which is a known and reported problem of the LMSR pricing[85]. We also note in Table 5.3 that the correlation coefficients between the data from the LMSR and BN models are fairly close to 1, revealing a strong correlation between them. Thus, the BN-based model is a realistic model of prediction markets.

q	z	**Correlation**
0.25	0.15	0.7925
0.25	0.5	0.7421
0.25	0.85	0.8556
0.55	0.15	0.8827
0.55	0.5	0.7912
0.55	0.85	0.7591
0.85	0.15	0.8404
0.85	0.5	0.8295
0.85	0.85	0.8661

Table 5.3: Correlations between the LMSR market price and the aggregated market price using BN from Figure 5.7. Observe that the numbers are fairly close to 1 which indicates a significant correlation between the LMSR and BN models.

5.2.4 Robustness to noise

It is known that real networks (biological/genetic, physical, neural, chemical, social, financial etc.) are always subject to disturbances and have the ability to reach functional diversity and aim to maintain the same state under environmental noise. Prediction markets can also be affected by some disturbances in the form of manipulation by the trading agents that reveal their beliefs untruthfully. For example, in the Tradesports 2004 presidential markets there was an apparent manipulation effort. An anonymous trader sold many securities corresponding to the event “George W. Bush will win the 2004 Presidential elections” at a very low value. This caused the market price to be driven to zero, implying a zero percent chance of the event happening. However, this manipulation effort failed, as the market price of the security related to this event rebounded rapidly to its previous level[110]. As prediction markets get more attention and become more widely known among the public,

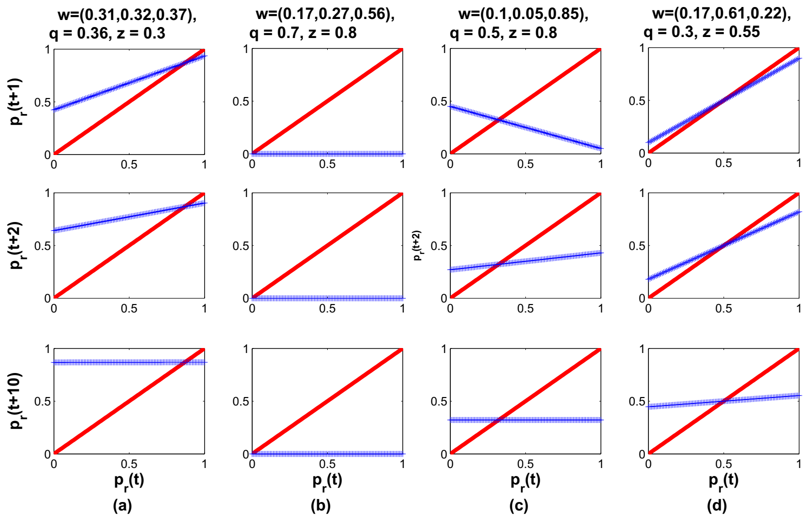


Figure 5.8: Iterations of the noisy aggregated market price (blue '+') with $j = 10$ versus the main diagonal (red line). We used the same parameters as in Figure 5.6.

it is likely that some individuals or groups will be motivated to manipulate them. Inducing disturbance in the system by changing the value of certain nodes in the network (according to a deterministic or stochastic rule) is a good model for an environmental or intrinsic type of perturbation. A similar procedure has been used for example by Bilke and Sjunnesson [7] where one randomly chosen variable is inverted after the system has reached a limit cycle in the Kauffman model, or by Goodrich and Matache[39] who show that the introduction of noise can stabilize a certain type of BN for a wide range of parameters. We analyze the response to disturbances of the prediction market in this chapter under a simple noise process to assess the robustness of the BN-based model to potential non-truthful trading agents.

We employ the following noise procedure, called the *"flip rule"*: at each trading period t we randomly select j trading agents and flip their state before applying the Boolean rule. This procedure has been used in[39]. Since the number of zeros and ones changes due to the perturbation, the value of $p_r(t)$ is modified prior to the application of the model (5.5). Now, if j nodes are chosen at random, then $j \cdot p_r(t)$ of them are in state 1 and $j \cdot (1 - p_r(t))$ are in state 0. By the flip rule, the total number of trading agents in state 1 is decreased by $j \cdot p_r(t)$ since they are changed to 0. On the other hand, the number is increased by $j \cdot (1 - p_r(t))$ since the zeros become ones. Thus, the proportion of trading agents in state 1, that is $p_r(t)$, becomes $p_r(t)\frac{j\cdot p_r(t)}{N} + \frac{j\cdot(1-p_r(t))}{N} = p_r(t) + \frac{j}{N}(1-2p_r(t))$. Clearly this number is in $[0,1]$. Then the formula (5.5) can be written as follows:

$$p_r(t+1) = \tag{5.7}$$

$$F_B\left(\frac{z - w_1(p_r(t) + \frac{j}{N}(1-2p_r(t)))}{w_3}\right)(1 - p_r(t) - \frac{j}{N}(1-2p_r(t)))$$

$$+F_B\left(\frac{z - w_1(p_r(t) + \frac{j}{N}(1-2p_r(t))) - w_2}{w_3}\right)$$

$(p_r(t) + \frac{j}{N}(1-2p_r(t)))$.

Figure 5.8 illustrates iteration plots analog to those in Figure 5.6 (for the same parameter values), but with induced perturbations. These results show that the noise generated by the "flip rule" can stabilize the prediction market as seen from the Figure 5.8(d). In that case the prediction market was chaotic without noise, and now it stabilizes around 0.5. This result supports the result obtained by Hanson[48], where he showed that the manipulator in the prediction market can aid its accuracy. For other parameter combinations, noise may change the value of the fixed points, maintaining stability, as can be seen in the other plots of Figure 5.8. The fixed points changed from 1 to 0.9 (Figure 5.8(a)), and from 0.5 to 0.4 (Figure 5.8(c)). In Figure 5.9 we show similar iteration plots but for parameter combinations that yield piecewise functions. We note that there may be multiple fixed points this time. However, all of them are stable since the derivative at those points is always less than 1. Therefore, the stability of the prediction market is either preserved or induced by the introduction of noise.

5.2.5 Scalability

In this section we test the scalability of our prediction market and analyze how the changes in the number of trading agents affect the dynamics of the prediction market. Figure 5.10 shows that our mean-field based model's accuracy for $p_r(t)$, the aggregated market price,

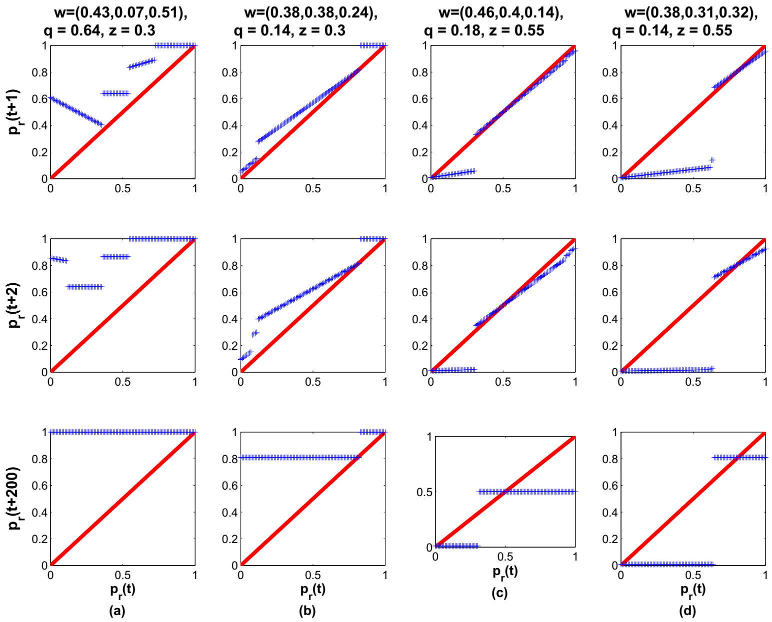


Figure 5.9: Iterations of the noisy aggregated market price (blue '+') with $j = 5$ versus the main diagonal (red line). Note that these parameter values yield piecewise functions with the possibility of multiple fixed points that are stable.

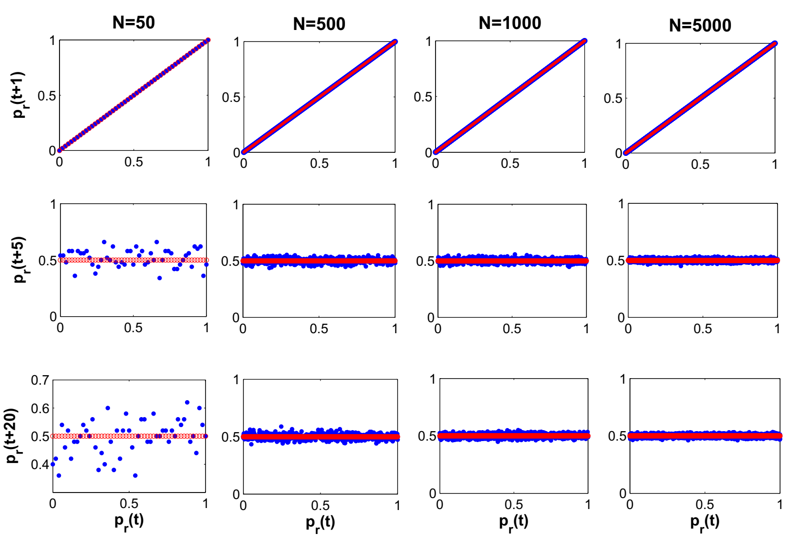


Figure 5.10: The system (blue dots) versus the mathematical model (red circles) under "flip" noise procedure for $50, 500, 1000, 5000$ trading agents for the 1st, 5th, and 20th trading periods. The parameters are fixed as follows: $w_1 = 0.1, w_2 = 0.05, w_3 = 0.85, z = 0.8, q = 0.5$.

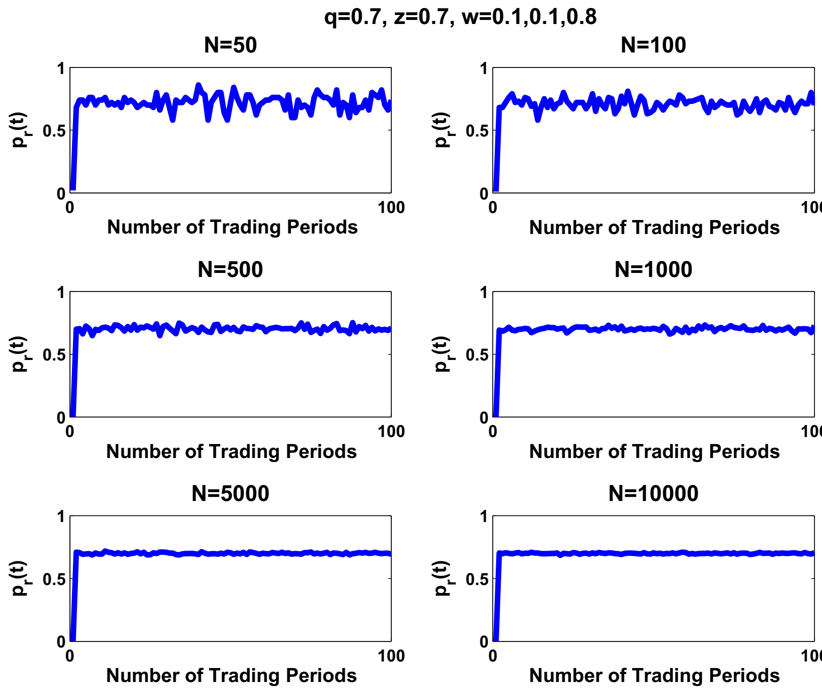


Figure 5.11: The aggregated market price by time steps, for different trader population sizes.

improves as the number of trading agents increases. This is expected, since a mean-field formula is valid in the limit $N \to \infty$. Figure 5.11 shows the aggregated market price using a BN for $50, 100, 500, 1000, 5000,$ and 10000 trading agents for $q = 0.7, z = 0.7, w_1, w_2, w_3 = 0.1, 0.1, 0.8$. This combination of z and q parameters yields a more dynamic behavior of the prediction market as seen in Figure 5.4 (c,d), however here we can see that as the number of trading agents increase the aggregated market price becomes less dynamic. However, there is not much difference in the market price dynamics when $N = 5000$ and when $N = 10000$, leading us to believe that in this case 5000 trading agents are enough to lead to an accurate prediction market.

In summary, in this chapter we showed that the BN approach gives results similar to the LMSR model with less fluctuations of the market price. In addition to proposing a new method to calculate the aggregated market price using BN and mean-field based mathematical modeling, we also showed how it can be used to analyze and predict the dynamics of the prediction market. The work presented in this chapter was published in[60] and the overview of our findings is shown in Table 5.4.

Contr.	Research Question	Research Findings
3	How does a prediction market evolve and what are its dynamics under different market and trader conditions?	- Our BN-based prediction market conforms to 4 distinct behaviors depending on parameters' values. - The market price can be used as a predictor for future market dynamics. - Our BN-based model can be used to determine how many traders are sufficient for an accurate prediction.
3	How can we make a prediction unaffected by "noise"?	- We find that within our BN-based model the noise does not decrease the accuracy of the market.

Table 5.4: Research findings for research questions in contribution 3 of this thesis.

Chapter 6

Information Aggregation for Multi-sensor Information Processing using Prediction Markets

In this chapter we tackle the problem of using prediction markets for decision making. Accurate information aggregation about uncertain events is very important for a decision maker. However, there has not been much effort in combining information aggregation and decision making. We consider the information aggregation problem with a decision maker in the multi-sensor domain. Multi-sensor fusion is concerned with the problem of fusing data from multiple sensors in order to make a more accurate estimation of the environment, and has been a central research topic in sensor-based systems[107]. Our work in this chapter is based on the insight that the problem addressed by prediction markets of aggregating the beliefs of different humans to forecast the outcome of an initially unknown event is analogous to the problem in multi-sensor fusion of fusing information from multiple sources to predict the outcome of an initially unknown object.

Recently several multi-agent techniques[105] have been proposed to address the multi-sensor fusion problem. Most of the solutions for multi-sensor information fusion and processing are based on Bayesian inference techniques[68;80;96]. While such techniques have been shown to be very effective, we investigate a complimentary problem where sensors can behave in a self-interested manner. Such self-interested behavior can be motivated by the sensors that deliberately misreport to save power, CPU cycles and memory (for example, to devote more resources to other tasks), and therefore always behave in a way that maximize their own benefits[72;95]. Moreover, these sensors may be programmed by different people and the assumption of full cooperation may not always hold[101]. To address this problem, we develop a multi-agent prediction market for multi-sensor information fusion that includes a utility driven mechanism to motivate each sensor, through its associated agent, to reveal accurate reports. Besides being an efficient aggregation mechanism, using prediction markets gives us several useful features - a mathematical formulation called a *scoring rule* that deters self-interested sensors from misreporting information, a regression-based belief update mechanism for the sensor agents for incorporating the aggregated beliefs (or information estimates) of other sensors into their own calculation, and the ability to incorporate an autonomous decision maker that uses expert-level domain knowledge to make utility maximizing decisions to deploy additional sensors appropriately to improve the detection of an object. Our experimental results illustrated with a landmine detection scenario while using identical data distributions and settings, show that the information fusion performed using our technique reduces the root mean squared error by $5 - 13\%$ as compared to a previously studied technique for landmine data fusion using the Dempster-Shafer theory[73] and by $3 - 8\%$ using distributed data fusion technique[70]. The research questions that we try to answer in this contribution are reproduced below in Table 6.1.

Contr.	Research Question	Research Approach
4	How can prediction markets be used for decision making?	Design and Development Theoretical Examination
4	Is there an advantage of using prediction markets for sensor fusion?	Empirical Analysis
4	Can a prediction market-based model for sensor fusion be effective in realistic settings given various limitations?	Empirical Analysis

Table 6.1: Research questions that are addressed in contribution 4 of this thesis.

6.1 Problem Formulation

To motivate our problem we describe a distributed automated landmine detection scenario used for humanitarian demining. An environment contains different buried objects, some of which could potentially be landmines. A set of robots, each equipped with one of three types of landmine detection sensor such as a metal detector (MD), or a ground penetrating radar (GPR) or an infra-red (IR) heat sensor, are deployed into this environment. Each robot is capable of perceiving certain features of a buried object through its sensor such as the object's metal content, area, burial depth, etc. However, the sensors give noisy readings for each perceived feature depending on the characteristics of the object as well as on the characteristics of the environment (e.g., moisture content, ambient temperature, sunlight, etc.). Consequently, a sensor that works well in one scenario, fails to detect landmines in a different scenario, and, instead of a single sensor, multiple sensors of different types, possibly with different detection accuracies can detect landmines with higher certainty[41]. Within this scenario, the central question that we intend to answer is: given an initial set of reports from *self-interested* sensors about the features of a buried object, what is a suitable set (number and type) of sensors to deploy over a certain time window to the object, so that, over this time window, the fused information from the different sensors successively reduces the uncertainty in determining the object's type.

Let L be a set of objects. Each object has certain features that determine its type. We assume that there are f different features and m different object types. Let $\Phi = \{\phi_1, \phi_2, ..., \phi_f\}$ denote the set of object features and $\Theta = \{\theta_1, \theta_2, ..., \theta_m\}$ denote the set of object types. The features of an object $l \in L$ is denoted by $l_\Phi \subseteq \Phi$ and its type is denoted by $l_\theta \in \Theta$. l_Φ can be perceived, albeit with measurement errors, through sensors, and, our objective is to determine l_θ as accurately as possible from the perceived but noisy values of l_Φ. Let $\Delta(\Theta) = \{(\delta(\theta_1), \delta(\theta_2), ..., \delta(\theta_m)) : \delta(\theta_i) \in [0,1], \sum_{i=1}^{m} \delta(\theta_i) = 1\}$, denote the set of probability distributions over the different object types. For convenience of analysis, we assume that when the actual type of object l, $l_\theta = \theta_j$, its (scalar) type is expanded into a m-dimensional probability vector using the function $vec : \Theta \to [0,1]^m : vec_j = 1, vec_{i \neq j} = 0$, which has 1 as its j-th component corresponding to l's type θ_j and 0 for all other components.

Let A denote a set of agents (sensors) and $A_{rep}^{t,l} \subseteq A$ denote the subset of agents that are able to perceive the object l's features on their sensors at time t. Based on the perceived object features, agent $a \in A_{rep}^{t,l}$ at time t reports a belief as a probability distribution over the set of object types, which is denoted as $\mathbf{b}^{a,t,l} \in \Delta(\Theta)$. The beliefs of all the agents are combined into a composite belief, $\mathbf{B}^{t,l} = Agg_{a \in A_{rep}^{t,l}}(\mathbf{b}^{a,t,l})$, and let $\hat{\Theta}^{t,l} : \mathbf{B}^{t,l} \to \Delta(\Theta)$ denote a function that computes a probability distribution over object types based on the aggregated agent beliefs. Within this setting we formulate the object classification problem as a decision making problem in the following manner: given an object l and an initial aggregated belief $\mathbf{B}^{t,l}$ calculated from one or more agent reports for that object, determine a set of additional agents (sensors) that need to be deployed at object l such that the following constraint is satisfied:

$$\min RMSE\left(\hat{\Theta}^{t,l}, vec(l_\theta)\right), \quad \text{for} \quad t = 1, 2, ...T, \tag{6.1}$$

where T is the time window for classifying an object l and RMSE is the root mean square error given by $RMSE(\mathbf{x}, \mathbf{y}) = \frac{||x-y||}{\sqrt{m}}$. In other words, at every time step t, the decision maker tries to select a subset of agents such that the root mean square error (RMSE) between the estimated type of object l and its actual type is successively minimized. The major components of the object classification problem described above consists of two parts: integrating the reports from the different sensors and making sensor deployment decisions based on those reports so that the objective function given in Equation 6.1 is satisfied. To address the first part, we have used distributed information aggregation with a multi-

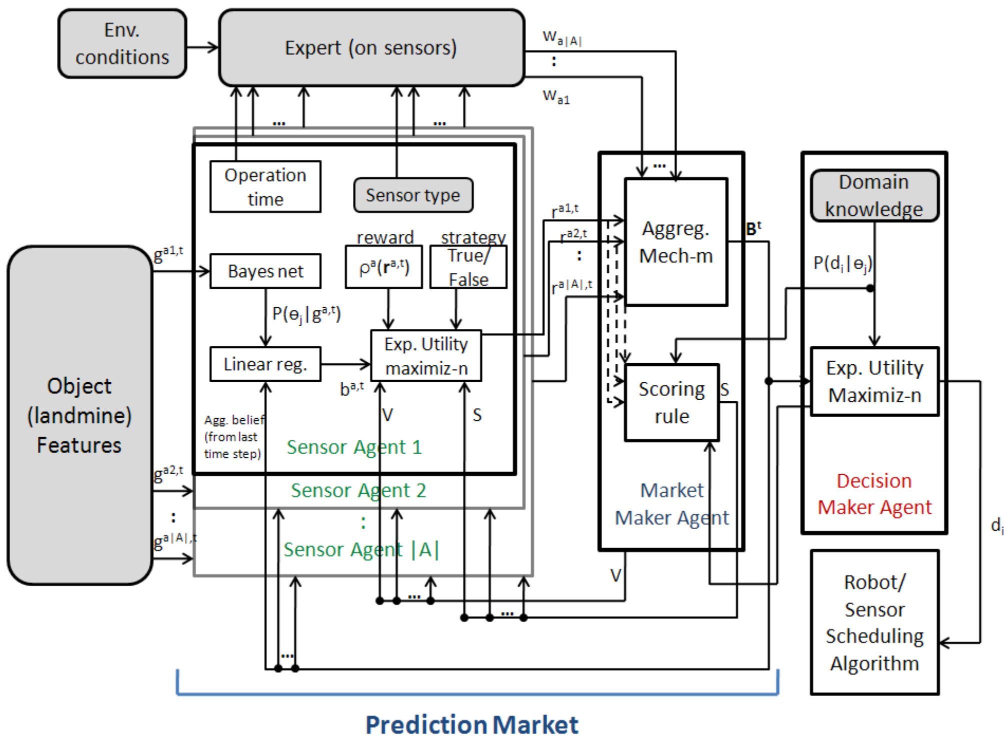


Figure 6.1: The different components of the prediction market for decision making and the interactions between them.

agent prediction market, while for the latter we have used an expected utility maximizing decision-making framework. A schematic showing the different components of our system and their interactions is shown in Figure 6.1 and explained in the following sections.

6.1.1 Sensor Agents

As mentioned in Section 6.1, there is a set of robots in the scenario and each robot has an on-board sensor for analyzing the objects in the scenario. Different robots can have different types of sensors and sensors of the same type can have different degrees of accuracy determined by their cost. Every sensor is associated with a software agent that runs on-board the robot and performs calculations related to the data sensed by the robot's sensor. In the rest of this chapter, we have used the terms sensor and agent interchangeably. For the ease of notation, we drop the subscript l corresponding to an object for the rest of this section. When an object is within the sensing range of a sensor (agent) a at time t, the sensor observes the object's features and its agent receives this observation in the form of an information signal $g^{a,t} =< g_1, ..., g_f >$ that is drawn from the space of information signals $G \subseteq \Delta(\Theta)$. The conditional probability distribution of object type θ_j given an information signal $g \in G$, $P(\theta_j|g) : G \to [0, 1]$, is constructed using domain knowledge[24;73;74] within a Bayesian network and is made available to each agent. Agent a then updates its belief distribution $\mathbf{b}^{a,t}$ using the following equation:

$$\mathbf{b}^{a,t} = w_{bel} \cdot \mathbf{P}(\Theta|g^{a,t}) + (1 - w_{bel}) \cdot \mathbf{B}^t, \tag{6.2}$$

where $\mathbf{B}^t$ is the belief value vector aggregated from all sensor reports.

Agent Rewards. Agents behave in a self-interested manner to ensure that they give their 'best' report using their available resources including sensor, battery power, etc. An agent a that submits a report at time t, uses its belief distribution $\mathbf{b}^{a,t}$ to calculate the report $\mathbf{r}^{a,t} =< r_1^{a,t}, ..., r_m^{a,t} >\in \Delta(\Theta)$. An agent can have two strategies to make this report - truthful or untruthful. If the agent is truthful, its report corresponds to its belief, i.e., $\mathbf{r}^{a,t} = \mathbf{b}^{a,t}$. But if it is untruthful, it deliberately reports an inaccurate belief to save its belief computation costs. Each agent a can update its report $\mathbf{r}^{a,t}$ within the time window T by obtaining new measurements from the object and using Equation 6.2 to update its belief. The report from an agent a at time t is analyzed by a human or agent expert[73] to assign a weight $w^{a,t}$ depending on the current environment conditions and agent a's sensor type's accuracy under those environment conditions (e.g., rainy weather reduces the weight assigned to the measurement from an IR heat sensor, or, soil that is high in metal content reduces the weight assigned to the measurement from an metal detector).

To motivate an agent to submit reports, an agent a gets an instantaneous reward, $\rho^{a,t}$, from the market maker for the report $\mathbf{r}^{a,t}$ it submits at time t, corresponding to its instantaneous utility, which is given by the following equation:

$$\rho^{a,t} = V(n^{t'=1..t}) - C^a(\mathbf{r}^{a,t}), \tag{6.3}$$

where $V(n^{t'=1..t})$ is the value for making a report with $n^{t'=1..t}$ being the number of times the agent a submitted a report up to time t, and, $C^a(\mathbf{r}^{a,t})$ is the cost of making report $\mathbf{r}^{a,t}$ for agent a. $C^a(\mathbf{r}^{a,t})$ is equal to the actual cost of resources used by the robot, such as expended time, battery power, etc. if the agent uses truthful strategy, and it is equal to some value $c_\epsilon <<$ actual cost of resources if the agent uses untruthful strategy. We denote the agent's value for each report $V(n^{t'=1..t})$ as a constant-valued function up to a certain threshold and a linearly decreasing function thereafter, to de-incentivize agents from making a large number of reports. Agent a's value function is given by the following equation:

$$V(n^{t'=1..t}) = \begin{cases} \nu & , n^{t'=1..t} \leq n^{threshold} \\ \frac{\nu(n^{t'=1..t} - n^{max})}{(n^{threshold} - n^{max})} & , otherwise \end{cases}$$

where $\nu \in \mathrm{Z}^+$, is a constant value that a gets by submitting reports up to a threshold, $n^{threshold}$ is the threshold corresponding to the number of reports a can submit before its report's value starts decreasing, and, n^{max} is the maximum number of reports agent a can submit before V becomes negative. Finally, to determine its strategy while submitting its report, an agent selects the strategy that maximizes its expected utility obtained from its cumulative reward given by Equation 6.3 plus an expected value of its final reward payment if it continues making similar reports up to the object's time window T.

6.1.2 Decision Maker Agent

The decision maker agent's task is to use the composite belief about an object's type, $\mathbf{B}^t$, given by the prediction market, and take actions to deploy additional robots(sensors) based on the value of the objective function given in Equation 6.1. Let AC denote a set of possible actions corresponding to deploying a certain number of robots, and $D = \{d_1, ...d_h\} : d_i \in Ac \subseteq AC$ denote the decision set of the decision maker, where h is the number of decisions the decision maker has. The decision function of the decision maker is given by $dec : \Delta(\Theta) \rightarrow D$. Let $\mathbf{u}_j^{dec} \in \mathrm{R}^m$ be the utility that the decision maker receives by determining an object to be of type θ_j and let $P(d_i|\theta_j)$ be the probability that the decision maker makes decision $d_i \in D$ given object type θ_j. $P(d_i|\theta_j)$ and $\mathbf{u}_j^{dec}$ are constructed using domain knowledge[24;73;74]. Given the aggregated belief distribution $\mathbf{B}^t$

at time t, the expected utility to the decision maker for taking decision d_i at time t is then $EU^{dec}(d_i, \mathbf{B}^t) = \sum_{j=1}^{m} P(d_i|\theta_j) \cdot \mathbf{u}_j^{dec} \cdot \mathbf{B}^t$. The decision that the decision maker takes at time t, also called its *decision rule*, is the one that maximizes its expected utility and is given by: $d^t = \arg\max_{d_i} EU^{dec}(d_i, \mathbf{B}^t)$.

6.2 Prediction Market

A conventional prediction market uses the aggregated beliefs of the market's participants or traders about the outcome of a future event, to predict the event's outcome. The outcome of an event is represented as a binary variable (event happens/does not happen). The traders observe information related to the event and report their beliefs, as probabilities about the event's outcome. The market maker aggregates the traders' beliefs and uses a scoring rule to determine a payment or payoff that will be received by each reporting trader. In our multi-agent prediction market, traders correspond to sensor agents, the market maker agent automates the calculations on behalf of the conventional market maker, and, an event in the conventional market corresponds to identifying the type of a detected object. The time window T over which an object is sensed is called the duration of the object in the market. This time window is divided into discrete time steps, $t = 1, 2, ..., T$. During each time step, each sensor agent observing the object submits a report about the object's type to the market maker agent. The market maker agent performs two functions with these reports. First, at each time step t, it aggregates the agent reports into an aggregated belief about the object, $\mathbf{B}^t \in \Delta(\Theta)$. Secondly, it calculates and distributes payments for the sensor agents. It pays an immediate but nominal reward to each agent for its report at time step t using Equation 6.3. Finally, at the end of the object's time window T, the market maker also gives a larger *payoff* to each agent that contributed towards classifying the object's type. The calculations and analysis related to these two functions of the market maker agent are described in the following sections.

Final Payoff Calculation. The payoff calculation for a sensor agent is performed by the market maker using a *decision scoring rule* at the end of the object's time window. A decision scoring rule[12] is defined as any real valued function that takes the agents' reported beliefs, the realized outcome and the decisions made by the decision maker as input, and produces a payoff for the agent for its reported beliefs, i.e. $S : \Delta(\Theta) \times \Theta \times D \longrightarrow \mathrm{R}$. We design a scoring rule for decision making that is based on how much agent a's final report helped the decision maker to make the right decisions throughout the duration of the prediction market and by how close the agent a's final report is to actual object type. Our proposed scoring rule for decision making given that object's true type is θ_j is given in Equation 6.4:

$$S(r_j^{a,t}, d^{[1:t]}, \theta_j) = \varpi(d^{[1:t]}, \theta_j) log \left(r_j^{a,t} \right), \tag{6.4}$$

where, $r_j^{a,t}$ is the reported belief that agent a submitted at time t for object type θ_j, $d^{[1:t]}$ is the set consisting of all the decisions that the decision maker took related to the object up to the current time t, θ_j is the object's true type that was revealed at the end the object's time window, $log \left(r_j^{a,t} \right)$ measures the goodness of the report at time t relative to the true object type θ_j, and, $\varpi(d^{[1:t]}, \theta_j)$ is the weight, representing how good all the decisions the decision maker took up to time t were compared to the true object type θ_j. $\varpi(d^{[1:t]}, \theta_j)$ is determined by the decision maker and made available to the agents through the market maker. We assume that $\varpi(d^{[1:t]}, \theta_j) = \sum_{i=1}^{t} P(d_i \mid \theta_j) \cdot \mathbf{u}_j^{dec}$, which gives the expected utility of the decision maker agent for making decision i when the true type of the object is θ_j.

Aggregation. Since a sensor agent gets paid both through its immediate rewards for making reports during the object's time window and through the scoring rule function for decision making at the end of the object's time window, we define the total payment that the agent has received by the end of the object's time window as a *payment function*.

Definition 1. *A function $\Psi(\mathbf{r}^{a,t}, d^{[1:t]}, \theta_j, n^{t'=1..t})$ is called a* ***payment function*** *if each agent a's total received payment at the end of the object's time window (when $t = T$) is*

$$\Psi(\mathbf{r}^{a,t}, d^{[1:t]}, \theta_j, n^{t'=1..t}) = \sum_{k=1}^{t} \rho^{a,k} + S(r_j^{a,t}, d^{[1:t]}, \theta_j) \tag{6.5}$$

where $\rho^{a,k}$, $S(r_j^{a,t}, d^{[1:t]}, \theta_j)$ and their components are defined as in Equations 6.3 and 6.4.

Let Ψ^{ave} denote a weighted average of the payment function in Equation 6.5 over all the reporting agents, using the report-weights assigned by the expert in Section 6.1.1, as given below:

$$\Psi^{ave}(\mathbf{r}^{A^t_{rep},t}, d^{[1:t]}, \theta_j, n^{A^t_{rep},t}) = \sum_{k=1}^{t} \sum_{a \in A^t_{rep}} w^{a,k} \rho^{a,k} + \varpi(d^{[1:t]}, \theta_j) \sum_{a \in A^t_{rep}} w^{a,t} log\left(r_j^{a,t}\right), \tag{6.6}$$

where A^t_{rep} is the subset of agents that are able to perceive object feature at time t and $w^{a,k}$ is the weight assigned to agent a at time k by the expert. To calculate an aggregated belief value in a prediction market, Hanson[47] used the generalized inverse function of the scoring rule. Likewise, we calculate the aggregated belief for our market maker agent by taking the generalized inverse of the average payment function given in Equation 6.6:

$$B^t_j = Agg_{a \in A^t_{rep}}(\mathbf{b}^{a,t}) = \frac{\frac{exp\left(\Psi^{ave} - \sum_{k=1}^{t} \sum_{a \in A^t_{rep}} w^{a,k}\rho^{a,k}\right)}{\varpi(d^{[1:t]}, \theta_j)}}{\sum_{\theta_j=\theta_1}^{\theta_m} \frac{exp\left(\Psi^{ave} - \sum_{k=1}^{t} \sum_{a \in A^t_{rep}} w^{a,k}\rho^{a,k}\right)}{\varpi(d^{[1:t]}, \theta_j)}} \tag{6.7}$$

where $B^t_j \in \mathbf{B}^t$ is the j-th component of the aggregated belief for object type θ_j. The aggregated belief vector, $\mathbf{B}^t$, calculated by the market maker agent is sent to the decision maker agent so that it can calculate its expected utility given in Section 6.1.2, as well as, sent back to each sensor agent that reported the object's type till time step t, so that the agent can refine its future reports, if any, using this aggregate of the reports from other agents. We also note that the aggregation mechanism used by our market maker is similar to an LMSR technique which was shown to lead to Weak Perfect Bayesian Equilibrium when, like in our setting, the information signals of the traders are independent conditional on the state of the world[18].

6.3 Payment function: Properties and Characteristics

In this section we first show that the payment function is proper, or incentive compatible. Then we show that when the market maker uses this payment function to reward each agent for its reported beliefs, reporting beliefs truthfully is the optimal strategy for each agent.

We can characterize a proper payment function similar to a proper scoring rule.

Definition 2. *A payment function* Ψ *is proper, or incentive compatible, if*

$$\Psi(\mathbf{b}^{\mathbf{a},\mathbf{t}}, d^{[1:t]}, \theta_j, n^{t'=1..t}) \geq \Psi(\mathbf{r}^{\mathbf{a},\mathbf{t}}, d^{[1:t]}, \theta_j, n^{t'=1..t}), \tag{6.8}$$

$\forall \mathbf{b}^{\mathbf{a},\mathbf{t}}, \mathbf{r}^{\mathbf{a},\mathbf{t}} \in \Delta(O)$. Ψ *is strictly proper if the inequality in Equation 6.8 strict.*

Payment functions can be shown to be proper by representing them using convex functions[12]. To show that our payment function in Equation 6.5 is proper, we characterize it in terms of a convex function, as shown below:

Theorem 2. *A payment function* Ψ *is proper for decision making if*

$$\Psi(\mathbf{r}^{\mathbf{a},\mathbf{t}}, d^{[1:t]}, \theta_j, n^{t'=1..t}) = G(\mathbf{r}^{\mathbf{a},\mathbf{t}}) - G'(\mathbf{r}^{\mathbf{a},\mathbf{t}}) \cdot (\mathbf{r}^{\mathbf{a},\mathbf{t}}) + \frac{G'_{i,j}(\mathbf{r}^{\mathbf{a},\mathbf{t}})}{P(d_i|\theta_j)}, \tag{6.9}$$

where $G(\mathbf{r}^{\mathbf{a},\mathbf{t}})$ *is a convex function and* $G'(\mathbf{r}^{\mathbf{a},\mathbf{t}})$ *is a subgradient of* G *at point* $\mathbf{r}^{\mathbf{a},\mathbf{t}}$ *and* $P(d_i|\theta_j) > 0$.

Proof. Consider a payment function Ψ satisfying Equation 6.9. We will show that Ψ must be proper for decision making. We will drop the agent and time subscripts in this proof, and also we will write $\Psi(\mathbf{r}, d^{[1:t]})$ (or its element $\Psi(r_j, d_i|\theta_j)$) instead of full $\Psi(\mathbf{r}^{\mathbf{a},\mathbf{t}}, d^{[1:t]}, \theta_j, n^{t'=1..t})$.

$$EU(\mathbf{b}, \mathbf{b}) = \sum_{i=1}^{h} \sum_{j=1}^{m} P(d_i|\theta_j) b_j \Psi(b_j, d_i|\theta_j) = \sum_{i=1}^{h} \sum_{j=1}^{m} P(d_i) b_j \left(G(\mathbf{b}) - G'(\mathbf{b}) \cdot \mathbf{b} + \frac{G'_{i,j}(\mathbf{b})}{P(d_i|\theta_j)} \right)$$

$$= G(\mathbf{b}) - G'(\mathbf{b}) \cdot \mathbf{b} + \sum_{i=1}^{h} \sum_{j=1}^{m} G'_{i,j}(\mathbf{b}) b_j = G(\mathbf{b}) - G'(\mathbf{b}) \cdot \mathbf{b} + G'(\mathbf{b}) \cdot \mathbf{b} = G(\mathbf{b}).$$

Since G is convex and G' is its subgradient, we have

$$EU(\mathbf{b}, \mathbf{r}) = \sum_{i=1}^{h} \sum_{j=1}^{m} P(d_i|\theta_j) b_j \Psi(r_j, d_i|\theta_j) = \sum_{i=1}^{h} \sum_{j=1}^{m} P(d_i|\theta_j) b_j \left(G(\mathbf{r}) - G'(\mathbf{r}) \cdot \mathbf{r} + \frac{G'_{i,j}(\mathbf{r})}{P(d_i|\theta_j)} \right)$$

$$= G(\mathbf{r}) - G'(\mathbf{r})(\mathbf{b} - \mathbf{r}) \leq G(\mathbf{b}) = EU(\mathbf{b}, \mathbf{b}).$$

Thus, Ψ is a proper payment function for decision making. Ψ is strictly proper payment function and the inequality is strict if G is a strictly convex function. $\square$

Proposition 2. *The payment function given in Equation 6.5 is proper.*

Proof. Let $G(\mathbf{b}) = EU(\mathbf{b}, \mathbf{b})$ and $G'_{i,j}(\mathbf{b}) = P(d_i|\theta_j)\Psi(\mathbf{b}, d^{[1:t]}, \theta_j, n^{t'=1..t})$. Then the payment function can be written in the form given in Equation 6.9 from Theorem 2. Therefore, the payment function Ψ given in Equation 6.5 is a proper payment function.

$$\Psi(\mathbf{b}, d^{[1:t]}, \theta_j, n^{t'=1..t}) = \sum_{i=1}^{h}\sum_{j=1}^{m} P(d_i|\theta_j) b_j \Psi(\mathbf{b}, d^{[1:t]}, \theta_j, n^{t'=1..t})$$
$$- \mathbf{b} \cdot \sum_{i=1}^{h}\sum_{j=1}^{m} P(d_i|\theta_j)\Psi(\mathbf{b}, d^{[1:t]}, \theta_j, n^{t'=1..t}) + \frac{\Psi(\mathbf{b}, d^{[1:t]}, \theta_j, n^{t'=1..t}) \cdot P(d_i|\theta_j)}{P(d_i|\theta_j)}$$
$$= \Psi(\mathbf{b}, d^{[1:t]}, \theta_j, n^{t'=1..t}).$$

$\square$

Agent Reporting Strategy. Assume that agent a's report at time t is its final report, then its utility function can be written as $u_j^{a,t} = \sum_{k=1}^{t} \rho^{a,k} + S(r_j^{a,t}, d^{[1:t]}, \theta_j)$. Then, agent a's expected utility for object type θ_j given its reported belief for object type θ_j, $r_j^{a,t}$, and its true belief about object type θ_j, $b_j^{a,t}$ at time t is

$$EU_j^a(r_j^{a,t}, b_j^{a,t}) = \sum_{i=1}^{h} P(d_i|\theta_j) b_j^{a,t} u_j^{a,t} = \sum_{i=1}^{h} P(d_i|\theta_j) b_j^{a,t} \left(\sum_{k=1}^{t} \rho^{a,k} + S(r_j^{a,t}, d^{[1:t]}, \theta_j) \right), \tag{6.10}$$

where $P(d_i|\theta_j)$ is the probability that the decision maker takes decision d_i when the object's type is θ_j.

Proposition 3. *If agent a is paid according to Ψ, then it reports its beliefs about the object types truthfully.*

Proof. The proof is a straight forward solution to expected utility maximization problem. Sensor agent a wants to maximize its expected utility function and solves the following program

$$\arg\max_{\mathbf{r}} \left(\sum_{i=1}^{h} \sum_{j=1}^{m} P(d_i|\theta_j) b_j^{a,t} \left[\sum_{k=1}^{t} \rho^{a,k} + \varpi(d^{[1:t]}, \theta_j) log\left(r_j^{a,t}\right) \right] \right),$$

s.t. $\sum_{j=1}^{m} r_j^{a,t} = 1$.

The Lagrangian is

$$\begin{aligned} L(\mathbf{r}, \lambda) = & \left(\sum_{i=1}^{h} \sum_{j=1}^{m} P(d_i|\theta_j) b_j^{a,t} \left[\sum_{k=1}^{t} \rho^{a,k} + \varpi(d^{[1:t]}, \theta_j) log\left(r_j^{a,t}\right) \right] \right) \\ & - \lambda \left(\sum_{j=1}^{m} r_j^{a,t} - 1 \right). \end{aligned}$$

The first order conditions are

$$\begin{aligned} & \frac{\partial L}{\partial r_j} = \sum_{i=1}^{h} \sum_{j=1}^{m} P(d_i|\theta_j) b_j^{a,t} \frac{\varpi(d^{[1:t]}, \theta_j)}{r_j^{a,*}} - \lambda = 0 \\ & \Rightarrow r_j^{a,*} = \frac{\varpi(d^{[1:t]}, \theta_j) b_j^{a,t} \sum_{i=1}^{h} P(d_i|\theta_j)}{\lambda} \\ & \frac{\partial L}{\partial \lambda} = -\sum_{j=1}^{m} r_j^{a,t} + 1 = 0. \end{aligned}$$

Substituting $r_j^{a,*}$ into the second equation above, we have

$$\begin{aligned} & \frac{\varpi(d^{[1:t]}, \theta_j) b_j^{a,t} \sum_{i=1}^{h} P(d_i|\theta_j)}{\lambda} = 1 \\ & \lambda = \varpi(d^{[1:t]}, \theta_j) \sum_{i=1}^{h} P(d_i|\theta_j) \\ & r_j^{a,*} = b_j^{a,t}. \end{aligned}$$

$\square$

```
selectStrategy()
foreach timestep t do
    if object is within sensing range of a then
        1. receive observation signals;
        2. update belief using Eqn. 6.2;
        3. calculate expected utility using Eqn. 6.10;
        4. choose report r^{a,t} that maximizes expected utility;
        5. send r^{a,t} to the decision maker;
        6. get instantaneous reward, ρ^{a,t};
    end
    else
        continue sensing;
    end
    observe the decision d^t made by the decision maker;
    get the aggregated belief distribution B^t from the market maker agent;
    if timestep t == object time window T then
        get final payoff;
    end
end
```

Algorithm 2: Algorithm used by agent a to select and submit reports

6.4 Experimental Results

We have conducted several experiments using our aggregation technique for decision-making within a multi-sensor landmine detection scenario. Our environment contains different buried objects, some of which are landmines. The true types of the objects are randomly determined at the beginning of the simulation. Due to the scarcity of real data related to landmine detection, we have used the domain knowledge that was reported in[24;73;74] to determine object types, object features, sensor agents' reporting costs, decision maker agent's decision set, decision maker agent's utility of determining objects of different types, and, to construct the probability distributions for $P(\theta_j|g)$ and $P(d_i|\theta_j)$. We report simulation results for root mean squared error (RMSE) defined in Section 6.1 and also for number of sensors over time, cost over object types, and average utility of the sensors over time.

Since the focus of our work is on the quality of information fusion, we will concentrate on describing the results for one object. We assume that there are robots with three types of sensors, MD (least operation cost, most noisy), IR (intermediate operation cost, moderately noisy), and GPR (expensive operation cost, most accurate). Initially, the object is detected

Decision	$P(d_i\|\theta_0)$	$P(d_i\|\theta_1)$	$P(d_i\|\theta_2)$
d_1(MD)	0.6	0.4	0
d_2(IR)	0.5	0.3	0.2
d_3(GPR)	0.4	0.3	0.3
d_4(MD,IR)	0.7	0.25	0.05
d_5(MD,GPR)	0.6	0.35	0.05
d_6(IR,GPR)	0.5	0.3	0.2
d_7(MD,MD,IR)	0.8	0.2	0
d_8(MD,MD,GPR)	0.75	0.2	0.05
d_9(IR,IR,MD)	0.7	0.3	0
d_{10}(IR,IR,GPR)	0.6	0.3	0.1
d_{11}(GPR,GPR,MD)	0.6	0.3	0.1
d_{12}(GPR,GPR,IR)	0.5	0.3	0.2
d_{13}(MD,IR,GPR)	0.9	0.1	0

Table 6.2: $P(d_i|\theta_j)$ values used for our simulation experiments.

Features	Meaning	Possible values	Sensors able to provide readings
$F1$	Metallic content	0(low), 1(high)	MD
$F2$	Area of the object	0(small), 1(large)	MD, GPR, IR
$F3$	Depth of the object	0(shallow), 1(deep)	MD, GPR
$F4$	Position of the sensor	0(near), 1(far)	MD, GPR, IR

Table 6.3: Object features used in our simulation experiments.

using one MD sensor. Once the object is detected for the first time, the time window in the prediction market for identifying the object's type starts. The MD sensor sends its report to the market maker in the prediction market and the decision maker makes its first decision based on this one report. We assume that decision maker's decision (sent to the robot/sensor scheduling algorithm in Figure 6.1) is how many $(0-3)$ and what type (MD,IR,GPR) of sensors to send to the site of the detected object subsequently. We have considered a set of 13 out of all the possible decisions under this setting as can be seen from Table 6.2. From[74], we derive four object features given in Table 6.3, which are metallic content, area of the object, depth of the object, and the position of the sensor. Combinations of the values of these four features constitute the signal set G and at each time step, a sensor perceiving the object receives a signal $g \in G$. The value of the signal also varies based on the robot/sensor's current position relative to the object. We assume that the identification of an object stops and the object type is revealed when either $\mathbf{B}_j^t \geq 0.95$, for any j, or after 10 time steps, or when there are no more sensors left. The default values for all domain related parameters are shown in Table 6.3 and the probability values, $P(\theta_j|g)$ and $P(d_i|\theta_j)$, are given in Tables 6.5 and 6.2 correspondingly. All of our results were averaged over 10 runs and the error bars indicate the standard deviation over the number of runs.

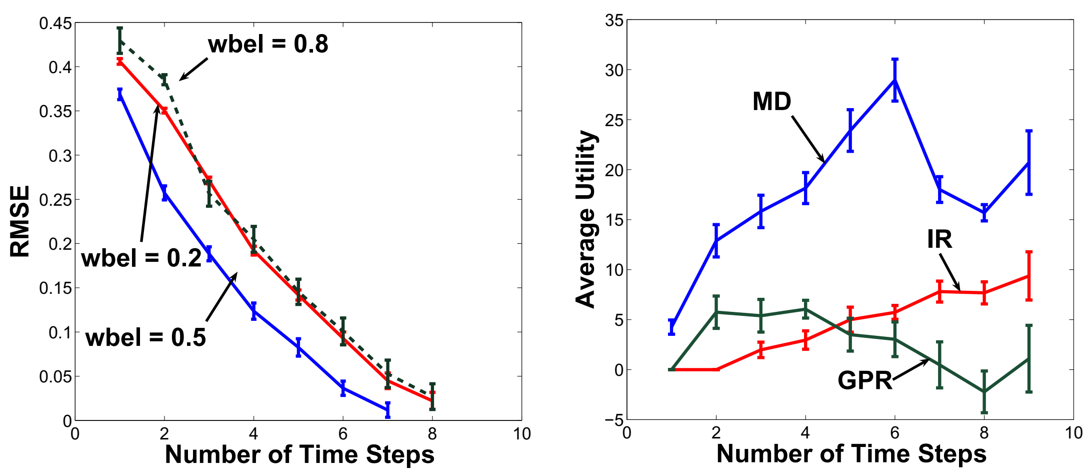
Name	**Value**
Object types	mine(θ_0), metallic object(θ_1)(non-mine), non-metallic object(θ_2)(non-mine)
Features	metallic content, object's area, object's depth, sensor's position
Sensor types	MD, IR, GPR
Max no. of sensors	10
Max no. of decisions	14
T (object identif-n window)	10
ν (agent's value if $n^{t'=1..t} \leq n^{threshold}$)	5
n_{max} (max no. of reports before value is negative)	20
$n_{threshold}$ (no. of reports before agent's value $< \nu$)	5

Table 6.4: Parameters used for our simulation experiments.

For our first group of experiments we analyze the performance of our technique w.r.t. the variables in our model, such as w_{bel} and time, and, w.r.t. to sensor and object types. We assume that there are a total of 5 MD sensors, 3 IR sensors, and 2 GPR sensors available to the decision maker for classifying this object. We observe that as more information gets sensed for the object, the RMSE value, shown in Figure 6.2(a), decreases over time. It takes on average $6-8$ time steps to predict the object type with 95% or greater accuracy depending on the object type and the value of w_{bel}. We also observe that our model performs the best with $w_{bel} = 0.5$ (in Equation 6.2), when the agent equally incorporates its private signal and also the market's aggregated belief at each time step into its own belief update. Figure 6.2(b) shows the average utility of the agents based on their type. We can see that

g	F1	F2	F3	F4	$P(\theta_0\|g)$	$P(\theta_1\|g)$	$P(\theta_2\|g)$
g_0	0	0	0	0	0.1	0.3	0.6
g_1	0	0	0	1	0.15	0.35	0.5
g_2	0	0	1	0	0.1	0.4	0.5
g_3	0	0	1	1	0.15	0.4	0.45
g_4	0	1	0	0	0.1	0.4	0.5
g_5	0	1	0	1	0.15	0.4	0.45
g_6	0	1	1	0	0.05	0.35	0.6
g_7	0	1	1	1	0.1	0.4	0.5
g_8	1	0	0	0	0.7	0.25	0.05
g_9	1	0	0	1	0.6	0.3	0.1
g_{10}	1	0	1	0	0.55	0.35	0.1
g_{11}	1	0	1	1	0.5	0.35	0.15
g_{12}	1	1	0	0	0.6	0.3	0.1
g_{13}	1	1	0	1	0.5	0.35	0.15
g_{14}	1	1	1	0	0.45	0.45	0.1
g_{15}	1	1	1	1	0.4	0.4	0.2

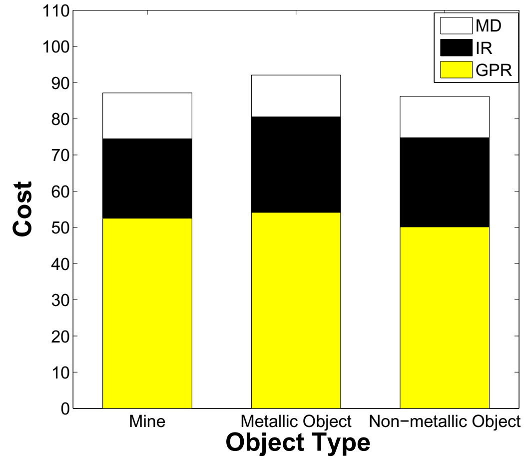
Table 6.5: $P(\theta|g)$ values used for our simulation experiments.



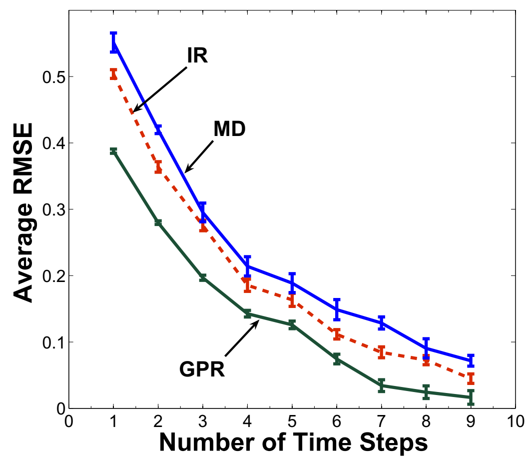
a b

Figure 6.2: RMSE for different values of w_{bel}(a), Average sensors' utilities for different sensor types(b).

MD sensors get more utility because their costs of calculating and submitting reports are generally less, whereas GPR sensors get the least utility because they encounter the highest cost. This result is further verified in Figure 6.3(a) where we can see the costs based on sensor types and also based on object types. We observe that detecting a metallic object



c



d

Figure 6.3: Cost for different object types(c), RMSE for sensors' reports averaged over sensor types(d).

that is not a mine has the highest cost. We posit that it is because both MD and IR sensors can detect metallic content in the object and extra cost is due to the time and effort spent differentiating metallic object from a mine. Although most of the mines are metallic[73;74], we can see that the cost of detecting a mine and a non-metallic object are similar because we require a prediction of at least 95%. Due to the sensitive nature of the landmine detection problem, it is important to ensure that even a non-metallic object is not a mine even if we encounter higher costs. However, despite MD's high utility (Figure 6.2(b)) and low cost (Figure 6.3(a)), its error of classifying the object type is the largest, as can be seen from Figure 6.3(b).

In our next group of experiments we analyze the effect of the total number of sensors that are available to the decision maker on the utility and the error using our prediction market-based technique. We keep all the parameters fixed as described in Table 6.3 except we vary the total number of sensors parameter. We also set the value of belief update weight $w_{bel} = 0.5$(used in Equation 6.2) and the object type to be a mine in these experiments. Figures 6.4 and 6.5 show the average utilities and average RMSE for different types of sensors. We observe that when there are diverse types of sensors available to the decision maker, the sensors get higher utility and the RMSE of detecting the object's type is lower

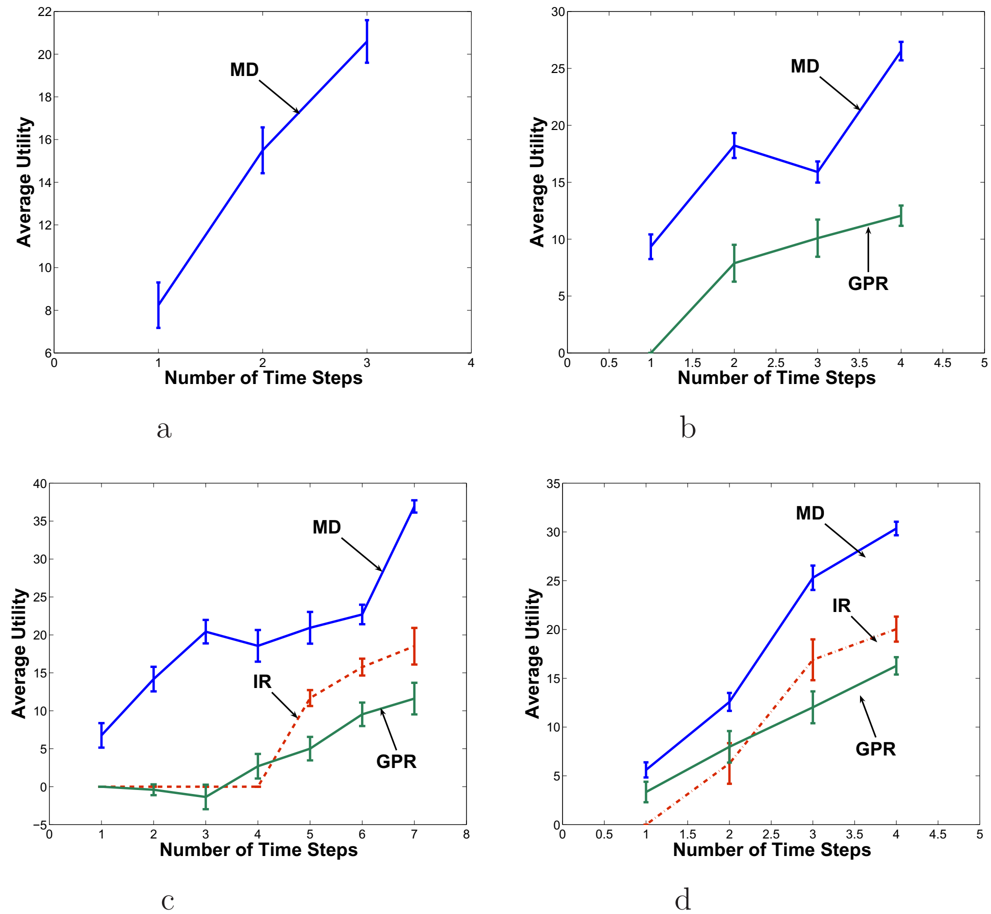


Figure 6.4: Average sensors' utilities in the environment with 5 MD sensors(a), 5 MD and 1 GPR sensor(b), 5 MD, 1 IR, and 1 GPR sensors(c), 2 MD, 2 IR, and 2 GPR sensors(d).

than when there are sensors of only one type available. For example, we can see that when the decision maker has only a total of 5 MD sensors available to it, MD sensors receive 32% less utility than when there are a total of 2 MD, 2 IR, and 2 GPR sensors available to the decision maker. We posit that this is because the sensors of different types sensor the environment differently and produce different beliefs. Thus, in the environment where there are sensors of different types, MD sensors take into account the beliefs of the sensors of other types through the market price and they are able to update their beliefs to reflect their private signals and also the beliefs of the sensors of other types. We also note that the accuracy of predicting object's type reaches only 80% when there are a total of 5 MD sensors

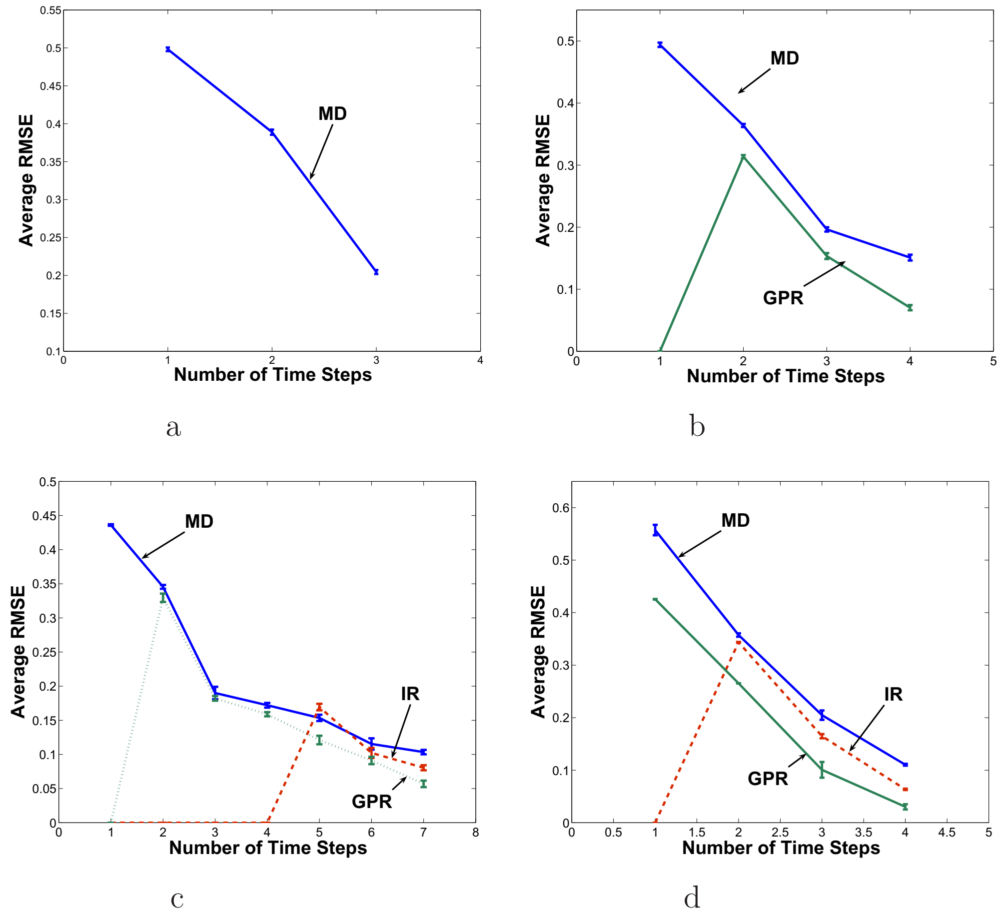
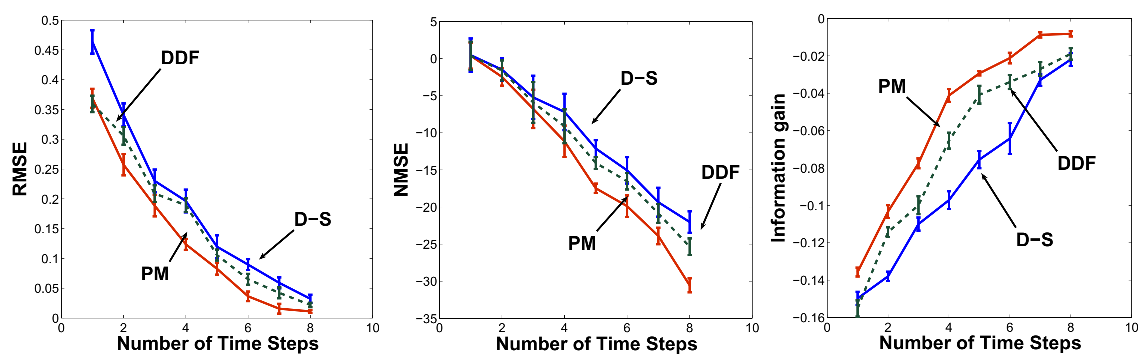


Figure 6.5: Average RMSE in the environment with 5 MD sensors(a), 5 MD and 1 GPR sensor(b), 5 MD, 1 IR, and 1 GPR sensors(c), 2 MD, 2 IR, and 2 GPR sensors(d).

available, but it reaches 94% when there are 2 MD, 2 IR, and 2 GPR sensors available. This is because when there are sensors of different types in the environment there is more diverse information available to the sensors, and also the decision maker has more opportunities to make better decisions.

Compared Techniques. For comparing the performance of our prediction market based object classification techniques, we have used two other well-known techniques for information fusion: (a) Dempster-Shafer (D-S) theory for landmine classification[73], where a two-level approach based on belief functions is used. At the first level, the detected object is classified according to its metal content. At the second level the chosen level of metal content

is further analyzed to classify the object as a landmine or a friendly object. The belief update of the sensors that we used for D-S method is the same one we have described in Section 6.1.1. (b) Distributed Data Fusion (DDF)[70], where sensor measurements are refined over successive observations using a temporal, Bayesian inference-based, information filter. To compare DDF with our prediction market-based technique, we replaced our belief aggregation mechanism given in Equation 6.7 with a DDF-based information filter. We compare our techniques using some standard evaluation metrics from multi-sensor information fusion[80]: root mean squared error (RMSE) defined as in Section 6.1, normed mean squared errors (NMSE) calculated as: $NMSE^t(\hat{\Theta}^t - vec(\theta_j)) = 10\, log_{10} \frac{\frac{1}{m}\sum_{j=1}^{m}\left(\hat{\Theta}_j^t, vec(\theta_j)\right)^2}{\frac{1}{m}\left(\sum_{j=1}^{m} vec(\theta_j)^2\right) - \left(\frac{1}{m}\sum_{j=1}^{m} vec(\theta_j)\right)^2}$, and, the information gain, also known as Kullback-Leibler divergence and relative entropy, calculated as: $D_{KL}^t(\hat{\Theta}^t || vec(\theta_j) = \sum_{j=1}^{m} \hat{\Theta}_j^t log\left(\frac{\hat{\Theta}_j^t}{vec(\theta_j)}\right)$.
$\hat{\Theta}^t$ was calculated using D-S, DDF, and our prediction market technique ($\hat{\Theta}^t = \mathbf{B}^t$).



a b c

Figure 6.6: Comparison of our Prediction Market-based information aggregation with Dempster-Shafer and Distributed Data Fusion Techniques using different metrics: RMSE(a), NMSE(b), Information gain(c).

In Table 6.6, we show how the decision maker's decisions using our prediction market technique results in the deployment of different numbers and types of sensors over the time window of the object. We report the results for the value of belief update weight $w_{bel} = 0.5$(used in Equation 6.2) while using our prediction market model, as well as using D-S and DDF. We see that non-metallic object classification requires less number of sensors as both MD and IR sensors can distinguish between metallic vs. non-metallic objects, and

Object type	**Time steps**	**PM**	**DDF**	**D-S**
Mine	1	1(1MD)	1(1MD)	1(1MD)
	2	3(1MD,1IR)	3(1MD,1GPR)	3(1IR,1GPR)
	3	4(1GPR)	5(1MD,1IR)	4(1MD)
	4	5(1MD)	6(1IR)	5(1MD)
	5	6(1MD)	7(1MD)	6(1IR)
	6	7(1IR)	8(1MD)	7(1IR)
	7	-	9(1IR)	8(1MD)
Metallic	1	1(1MD)	1(1MD)	1(1MD)
or	2	3(1MD,1IR)	4(1MD,1IR,1GPR)	3(1IR,1GPR)
Friendly	3	4(1GPR)	5(1MD)	4(1MD)
for D-S	4	5(1MD)	6(1IR)	5(1IR)
	5	6(1IR)	7(1MD)	6(1IR)
	6	7(1MD)	8(1IR)	7(1MD)
	7	8(1IR)	9(1GPR)	8(1MD)
	8	-	9(1MD)	8(1MD)
Non-	1	1(1MD)	1(1MD)	
metallic	2	2(1MD)	2(1IR)	
	3	3(1IR)	3(1MD)	
	4	4(1MD)	4(1GPR)	
	5	5(1IR)	5(1MD)	
	6	6(1MD)	6(1IR)	
	7	-	7(1MD)	

Table 6.6: Different number of sensors and the sensor types deployed over time by a decision maker to classify different types of objects.

so, deploying just these two types of sensors can help to infer that the object is not a mine. In contrast, metallic objects require more time to get classified as not being a mine because more object features using all three sensor types need to be observed. We also observe that on average our aggregation technique using prediction market deploys a total of $6-8$ sensors and detects the object type with at least 95% accuracy in $6-7$ time steps, while the next best compared DDF technique deploys a total of $7-9$ sensors and detects the object type with at least 95% accuracy in $7-8$ time steps.

Our results shown in Figure 6.6(a) illustrate that the RMSE using our PM-based technique is below the RMSEs using D-S and DDF by an average of 8% and 5% respectively. Figure 6.6(b) shows that the NMSE values using our PM-based technique is 18% and 23%

Contr.	Research Question	Research Findings
4	How can prediction markets be used for decision making?	- With the design of the payment function that incentivizes truthful revelation.
4	Is there an advantage of using prediction markets for sensor fusion?	- Our results showed that PM-based technique for sensor fusion results in more accurate results than existing methods.
4	Can a prediction market-based model for sensor fusion be effective in realistic settings given various limitations?	- Our simulation results show good results for a realistic scenario in a sensor fusion setting.

Table 6.7: Research findings for research questions in contribution 4 of this thesis.

less on average than D-S and DDF techniques respectively. Finally, in Figure 6.6(c) we observe that the information gain for our PM-based technique is 12% and 17% more than D-S and DDF methods respectively.

In summary, in this chapter of the thesis we described a sensor information aggregation technique for object classification with a multi-agent prediction market and developed a payment function used by the market maker to incentivize truthful revelation by each agent. Our experimental results verify that, for identical data distributions and settings, using our prediction market-based information aggregation technique increases the accuracy of object classification favorably as compared to two other commonly used techniques. The work presented in this part of the thesis was published in[57] and[58] and the overview of our findings is shown in Table 6.7.

Chapter 7

Weighted Bayesian Graphical Games for Distributed Prediction Markets

In the last part of this thesis we investigate a setting of several prediction markets that are able to interact and influence each other's aggregated market prices, and thus the predictions about the outcome of future events. We call such setting *distributed prediction markets*. Contrary to single prediction markets with confined traders, there are several real-life instances where multiple prediction markets running simultaneously have similar events. For example, both Intrade and Iowa Electronic Market ran prediction markets on several events related to the 2012 U.S. Presidential elections. With such similar events across markets, it is very likely that the expected outcomes (prices) of an event in one market will influence the price of the same or similar event in a different market. There are several ways in which prediction markets can influence each other. For example, traders participating in prediction markets, just like in financial markets, are not locked in one prediction market; they can trade in other markets. In such case, the market participants' trading decisions maybe affected by their involvement in other related markets. Also, traders can observe the fluctuations in the market prices of all related prediction markets, and thus their trading decisions maybe influenced by that information. Such inter-market influence is frequently observed in financial markets, which operate very similarly to prediction markets. For example, after analyzing the data from the retail online brokerage, called eToro, Pan *et al.* reported that the prices of securities in one market affect the prices of similar securities in other markets and that social trading (when traders can see each others' trades) results in higher profits to the traders[88]. Financial market traders also tend to exhibit herding behavior[3], where a trader tends to have the same beliefs as the majority of the traders in its neighborhood. For example, a trader is more prone to riskier behavior (e.g., overreacting to small price changes) when its peer traders exhibit risky behavior. Since a market maker in a prediction market can be viewed as type of trader, they can also influence each other in the similar manner. Therefore, it is important to consider the global setting, where the aggregated market price in one prediction market is not only affected by the traders in that market, but also by the expected outcome of similar events in other prediction markets. Furthermore, we envisage that by taking into account the aggregated information of other prediction markets, the aggregation of each prediction market can be improved. In this chapter we attempt to make a first step into the study of distributed prediction markets by investigating the effects of events across multiple prediction markets and analyze how prices evolve due to such inter-market effects.

The main contributions made by this part of the thesis towards studying this problem are the following: we describe a model of a distributed prediction market that comprises multiple, parallel running prediction markets and uses a graphical structure between the market makers of the different markets to represent inter-market influence. We then propose a formal framework based on graphical games[64] called a Weighted Bayesian Graphical Game (WBGG) to capture the interaction between multiple market makers and describe an algorithm based on NashProp[79] to calculate an approximate Bayes-Nash equilibrium efficiently for an n-agent WBGG. Finally, we conduct a number of experiments to analyze the effect of different parameters in distributed prediction markets, and we find that when the size of the neighborhood is too small or too large, agents' utilities decrease, and, that agents using our algorithm in a distributed prediction market setting can outperform the agents using a greedy strategy or the agents in the setting where prediction markets are disjoint. To the best of our knowledge, this work represents the first attempt at studying inter-market influences between similar events across multiple prediction markets through strategic decision making by market makers. The research questions that we try to answer in this contribution are reproduced below in Table 7.1.

Contr.	Research Question	Research Approach
5	How can we design a distributed model of a prediction market to be used for decision making?	Theoretical Examination Design and Development
5	Is our distributed model of the prediction market incentive compatible?	Theoretical Examination
5	How does a distributed information aggregation in a prediction market compare to the centralized aggregation?	Theoretical Examination Empirical Analysis

Table 7.1: Research questions that are addressed in contribution 5 of this thesis.

7.1 Distributed Prediction Markets

In this section we define and characterize distributed prediction markets. We consider n prediction markets with each prediction market having one market maker that is responsible for aggregating traders' beliefs and setting the market price for its market. The events across the prediction markets can be correlated and the market makers can interact with each other if their prediction markets are running correlated events. Let $N = \{1, ..., n\}$ denote the set of market makers with i being the market maker for the i-th prediction market. Let $\Gamma = \{\Gamma_1, ..., \Gamma_n\}$ denote a set of trading agents with Γ_i being the set of trading agents in the i-th prediction market. Note that trading agents can participate in multiple prediction markets simultaneously, i.e., $\Gamma_i \bigcap \Gamma_j \neq \emptyset$. Also let $E = \{E_1, ..., E_n\}$ denote the set of events across all the prediction markets with E_i representing the set of events in the

i-th prediction market*. Finally, let P denote the trader population.

Next, we present two axioms that outline the behavior of market makers in the distributed prediction market setting. Consider two market makers i and j running events e_{f_i} and e_{g_j} in their respective prediction markets. Let d_{f_i,g_j} be a distance metric that measures the similarity between the definitions of the two events e_{f_i} and e_{g_j} †.

Definition 3. Related Event. *An event e_{f_i} is related to event e_{g_j} if $d_{f_i,g_j} > \epsilon_{sim}$, where ϵ_{sim} is a constant.*

We denote the number of market makers that market maker i interacts with by η_i and the influence of the market maker i on market maker j by ϖ_{ij}.

Axiom 1. ***Local interaction, Influence, Competition:*** *If events e_{f_i} and e_{g_j} run by market makers i and j correspondingly are related, then (1) market maker i* interacts *with market maker j for determining the price of event e_{f_i}, (2) $0 < \varpi_{i,j} < 1$, and (3) market makers i and j are competitive.*

The first part of Axiom 1 determines the criterion for interaction between two market makers. Market maker i interacts with another market maker j for updating event e_{f_i}'s price, only if j's market has an event that is related to event e_{f_i}. The second part of Axiom 1 states that two market makers that have a pair of related events in their market have a non-zero, positive influence on each other. Influence values are normalized to a range of 0 and 1. In a prediction market a market maker needs to be able to calculate the market price (aggregate) and to stimulate the trading by always allowing traders to buy or sell securities. If there are prediction markets with similar events, the traders may choose one prediction market over the other, and market makers may end up competing over the traders just like in financial markets[61]. The third part of Axiom 1 summarizes this competitive behavior between market makers.

*For legibility, we refer to the security corresponding to an event as the event itself.

†We assume that d_{f_i,g_j} is based on the similarity between the textural description of the events e_{f_i} and e_{g_i} and is provided externally to the market makers either by a human expert or by an automated program.

Axiom 2. ***Incentives:*** *Let θ_i be the private information of market maker i denoted as i's type. Let $u_i(\theta_i)$ be the utility that the market maker i gets for interacting with other market makers when its type is θ_i. If $d_{f_i,g_j} > \epsilon_{sim}$, for any i, j, e_{f_i}, e_{g_j}, then $\exists\ \theta'_i$ such that $u_i(\theta'_i) > u_i(\theta_i)$.*

Finally, because market makers are competing with each other to attract traders on related events, a market maker may have incentives to misreport its inside information about aggregated prices when interacting with other market makers. In other words, market-makers have preferences over their types to improve their utility as mentioned in Axiom 2.

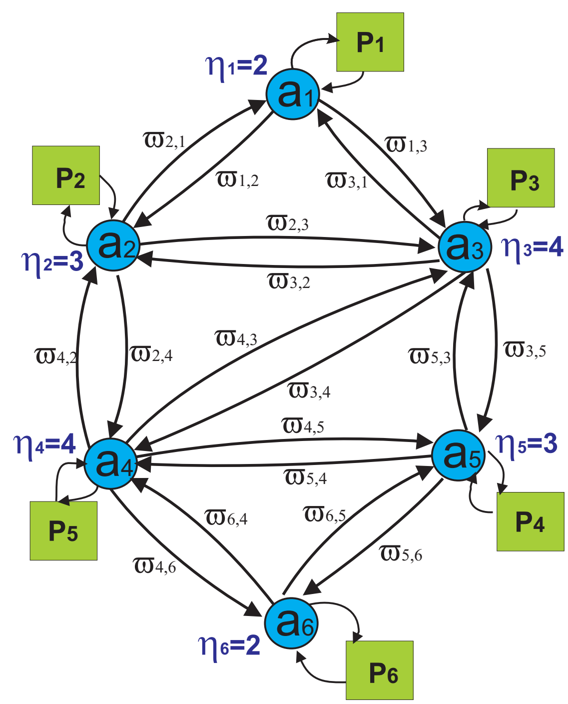


Figure 7.1: Example of the distributed prediction markets with 6 market makers $a_1, ..., a_6$.

Definition 4. Distributed prediction market. *A distributed prediction market is specified by the tuple $\mathcal{M} = \langle N, P, E, \mathcal{W}\rangle$ where N, P, E are as defined before and $\mathcal{W} = \{\varpi_{ij} : i, j \in N\}$.*

Figure 7.1 shows a diagram of a distributed prediction market with 6 market makers $a_1, ..., a_6$, and corresponding trader populations $P_1, ..., P_6$. η_i denotes the number of agents

a_i interacts with. For the simplicity of the analysis in the rest of the chapter we assume the setting where multiple market makers interact over one related event; but the results are valid for multiple events. Since the decision making for inter-market influence is done mainly by market makers, we abstract the operation of the trading agents and assume that the intra-market price of each prediction market is updated by the market maker's actions in the locality of the current market price. Nevertheless, our proposed technique can be combined easily with any other intra-market price update method such market scoring rules. Since we are not focussing on the interactions of trading agents, we refer to market makers as agents in the rest of the chapter.

7.1.1 Graphical Games for Distributed Prediction Markets

We propose a form of graphical games[64;79;100] as a formal model for the interaction between market makers in a distributed prediction market. *Graphical games* are a compact representation of complete information, one-shot, normal-form games that use graphical models to represent the set of agents whose actions influence each others' payoffs. A graphical game representation is appropriate for modeling distributed prediction markets because they can capture the interactions and influences between agents that are within a certain local neighborhood of each other, unlike for example conventional Bayesian games. A graphical game is described by an undirected graph G in which agents are identified with nodes, and the edge between two nodes implies that the payoff of each of the two agents is dependent on the other agent's actions. Every graphical game has a Nash equilibrium and every game can be represented as a graphical game by letting the game to be the complete graph. Representation of graphical games is $O(n2^k)$ with $k = max_i \eta_i \ll n$, i.e. it is exponential in the maximum degree of any node in the graph in comparison to normal form games which representation is $O(n2^n)$, i.e. exponential in the number of agents[64;79].

The original work by Kearns et al.[64] considered acyclic graphical games of complete information in which the underlying graph is a tree and presented a message-passing algorithm, known as TreeProp or KLS algorithm, for computing approximate Nash equilibria (NE) efficiently. The main idea of their algorithm is to view the game as being composed of several interacting local games and to exploit this locality by iteratively computing local equilibria and combining them together to obtain global equilibria efficiently. Consequently, in[79] the authors generalized the TreeProp algorithm to an arbitrary graph structure by proposing a message-passing NashProp algorithm for complete information games which involves an approach analogous to loopy belief propagation in graphical games. In other work, Vickrey and Koller[104] presented multi-agent algorithms for solving graphical games including hill-climbing, constraint satisfaction, and hybrid approaches. In[100] the authors studied the graphical games with incomplete information with discrete and continuous types and propose an extension to KLS algorithm to find Bayes-Nash equilibrium efficiently in a tree structured graphical game. In games of incomplete information, the payoff to an agent depends not only on the actions of the other agents but also on its own private type. Games represented as sparsely connected graphs can be commonly be seen in various settings, for example social networks, business relationships, financial markets, ad hoc networks of mobile devices, etc. Kearns[63] has also argued that graphical games provide computational, structural and interdisciplinary advantages as well.

We present an augmented form of the conventional graphical game to represent distributed prediction markets, called a Weighted Bayesian Graphical Game (WBGG). Unlike previous works on graphical games, WBGG incorporates incomplete information, the influence of agents on each other as a pair of *directed* edges, and an arbitrary graphical structure in one representation. We define a WBGG as follows:

Definition 5. *A WBGG is a tuple* $(N, \Theta, p, \Xi, \mathcal{W}, A, u)$*, where*

- $N = \{1, ..., n\}$ *- set of market maker agents.*
- $\Theta = \Theta_1 \times ... \times \Theta_n$*, where* Θ_i *is the type space of agent* i*.*
- $p : \Theta \rightarrow [0, 1]$ *is the common prior over types.*
- $\Xi = \{\Xi_1, ..., \Xi_n\}$ *- set of directed edges, where* $\Xi_i = \{\xi_{ij} | \xi_{ij} = (i, j), i, j \in N\}$ *with* ξ_{ij} *being an edge between agents* i *and* j *that are able to interact.*

- $\mathcal{W}$ - *set of edge weights, where* $\varpi_{ij} \in \mathcal{W}$ *is the weight of the edge* ξ_{ij} *between agents* i *and* j. ϖ_{ij} *is determined by agent* i *and indicates the influence of agent* i *on agent* j. *Given agent* i*'s type is* θ_{iopt}, ϖ_{ij} *is calculated as:*

$$\varpi_{ij} = \alpha p(\theta_{jopt}|\theta_{iopt}) + (1-\alpha)p(\theta_{jpes}|\theta_{iopt}), \tag{7.1}$$

where $\theta_{iopt} \in \Theta_i$, $\theta_{jopt}, \theta_{jpes} \in \Theta_j$ *and* α *is a confidence parameter representing* i*'s belief that* j *is of the same type as itself.*

- $A = A_1 \times ... \times A_n$, *where* A_i *is a finite set of actions available to agent* i.

- $u_i : \Theta_{i \in N} \times A_{i \in N} \times T \rightarrow R$ *is the utility of agent* i.

We assume that a set of similar events in a distributed prediction market have a duration of T periods with t denoting the current time period. The agents that are able to interact and therefore influence each other's utilities define a neighborhood $\mathcal{N}$, where $\mathcal{N}_{-i} = \{j | j \in N_{-i}, \xi_{ij} \in \Xi_i\}$ and $\mathcal{N}_i = \mathcal{N}_{-i} \cup \{i\}$. We also assume that each agent can be one of two possible types, i.e. if Θ_i is the type space of market maker agent i then $\Theta_i = \{\theta_{iopt}, \theta_{ipes}\}$ with θ_{iopt} implying that agent i is an *optimistic* market maker agent and θ_{ipes} implying that it is a *pessimistic* market maker agent.

Following Bayesian games[76] we use $s_i(\theta_{iopt})$ to denote agent i's mixed strategy over A_i given its type is θ_{iopt}. S_i is the set of all i's mixed strategies. We use notation s_i for unconditional mixed strategy of agent i. We use $A_{\mathcal{N}_i}, \overline{s}_{\mathcal{N}_i}, \overline{\theta}_{\mathcal{N}_i}$ to denote the vector of actions, strategies and types of all agents in the neighborhood of agent i, $\overline{\varpi}_{\mathcal{N}_{-i}}$ denote the vector of weights between agent i and all agents in the neighborhood of agent i, a_i, s_i, θ_i to denote the action, strategy and type of agent i, and $A_{\mathcal{N}_{-i}}, \overline{s}_{\mathcal{N}_{-i}}, \overline{\theta}_{\mathcal{N}_{-i}}$ to denote actions, strategies and types of all agents in the neighborhood of agent i except agent i itself.

Agent action set and utility function. For specifying the actions in agent i's action set A_i, we assume that agent i can have two possible actions in A_i - to raise the current market price or to lower it by a certain amount that is specified by a *jump* parameter λ_i^t, i.e., $A_i = \{\lambda_i^t, -\lambda_i^t\}$. To prevent arbitrary values of λ_i^t, we make λ_i^t inversely proportional to the market price π_i^t, i.e. $\lambda_i^t = \frac{\delta_i^t}{\pi_i^t}$, where $\pi_i^t \in (\zeta, 1]$ is the market price at time step t, with ζ being a small positive constant corresponding to a very nominal price change, and δ_i^t is a constant that determines the direction (up or down) of the price change *. The value of δ_i^t is determined by agent i depending on its type and by observing the direction of the market prices changes among its neighbors, as given by the following equations:

$$\delta_{i|\theta_i=opt}^t = \begin{cases} +\zeta & \text{if } \frac{\sum_{j\in\mathcal{N}_{-i}} \pi_j^t - \pi_j^{t-1}}{|\mathcal{N}_{-i}|} \geq 0, \\ -\zeta & \text{otherwise} \end{cases}$$

$$\delta_{i|\theta_i=pes}^t = \begin{cases} +\zeta & \text{if } \forall j \in \mathcal{N}_{-i},\, \pi_j^t - \pi_j^{t-1} \geq 0, \\ -\zeta & \text{if } \exists j \text{ s.t. } \pi_j^t - \pi_j^{t-1} < 0 \end{cases}$$

If agent i is optimistic, it sets $\delta_i^t = +\zeta$ if the average change in the market price of its neighbors in the last time step has been non-negative, otherwise it sets $\delta_i^t = -\zeta_i$. Similarly, if agent i is pessimistic, it sets $\delta_i^t = +\zeta_i$ only if all of its neighbors increased their prices in the last time step but sets $\delta_i^t = -\zeta_i$ if at least one of its neighbors decreased its prices in the last time step. †

The utility of agent i is calculated as:

$$u_i^t(\lambda_i) = (T-t)e^{-\lambda_i(T-t)}, \tag{7.2}$$

where t is the current time period ($t = 0$ at the start of the market). The above utility equation guarantees that the utility of changing the market price (by taking an action in the WBGG) is proportional to the remaining duration of the event in the market, and, more exploration (large price changes) gives higher utility towards the beginning of the event, but as the event nears its end and its price converges, large explorations are punished with lower utility.

*If π_i crosses either its lower or upper bound due to action A_i, we set it back to its lower or upper bound correspondingly.

†We drop superscript t from λ_i^t, δ_i^t, and π_i^t henceforth, assuming it to be understood from the context.

Next, we define the agent's expected utility in a weighted Bayesian graphical game as

$$EU_i(s_i, \overline{s}_{\mathcal{N}_{-i}}, \theta_i) = \sum_{a_{\mathcal{N}_i} \in A_{\mathcal{N}_i}} \overline{\varpi}_{\mathcal{N}_{-i}} \times \left(s_i(a_i) \overline{s}_{\mathcal{N}_{-i}}(A_{\mathcal{N}_{-i}} | \overline{\theta}_{\mathcal{N}_{-i}}) u_i(A_{\mathcal{N}_i}, \overline{\theta}_{\mathcal{N}_{-i}}, \theta_i) \right), \tag{7.3}$$

where $\overline{\varpi}_{\mathcal{N}_{-i}} = \prod_{j \in \mathcal{N}_{-i}} \varpi_j$. Note that agent i has to consider every assignment of types to the other agents in its neighborhood $\theta_{\mathcal{N}_{-i}}$ and every action profile $\overline{a}_{\mathcal{N}_i}$ in order to calculate the utility $u_i(\overline{a}_{\mathcal{N}_i}, \overline{\theta}_{\mathcal{N}_{-i}}, \theta_i)$.

Definition 6. *For agent i a strategy s_i is said to be* ***best response (BR)*** *in a WBGG for type θ_i to $\overline{\theta}_{\mathcal{N}_i}$, if*

$$\forall s', EU(s_i, \overline{s}_{\mathcal{N}_{-i}}, \theta_i) \geq EU(s', \overline{s}_{\mathcal{N}_{-i}}, \theta_i) \tag{7.4}$$

Definition 7. *A strategy vector $\overline{s}$ is a* ***Bayes-Nash Equilibrium (BNE)*** *in a WBGG if and only if every agent i is playing a best response to the others.*

7.1.2 Computing Bayes-Nash Equilibrium

In this section we first present an abstract algorithm for computing BNE in a weighted Bayesian graphical game of an arbitrary graphical structure. This algorithm is similar to NashProp algorithm[79] that has been extended to incomplete games with an arbitrary graphical structure, thus the definition of the expected utility and the best response have been modified.

For now, we will not purposefully specify a certain representation and a certain implementation. After proving the correctness of the abstract algorithm, we will fill in the unspecified gaps. The abstract algorithm is basically a two-stage message passing algorithm. In the first step, local optimal response is found for each agent, where each agent calculates the optimal strategy given its neighbor's strategies and sends it to its neighbors. In the second step, global solution is constructed by eliminating inconsistent local optimal response.

Let $D_{i,j}$ be the binary table indexed by all possible strategies of agent i and agent j

that is sent from agent j to agent i. Let $\mathcal{N}_{-i}^{j} = (1, ..., m-1)$ denote the neighbors of agent j besides agent i. And let $\overline{s}_{\mathcal{N}_{-i}^{j}} = (s_1, ..., s_{m-1})$ be the vector of mixed strategies of agents in $\mathcal{N}_{-i}^{j}$, called the *witness* to $D_{i,j}$. Also let P_j be the projection set used to combine information sent to agent j by its neighbors.

Theorem 3. *Algorithm 3 computes BNE for an arbitrary graphical game and the tables and witnesses calculated by it contain all possible BNE of the game.*

Proof. The proof is a constructive argument of the workings of the algorithm. The stage 1 starts with an arbitrary node. Each node(agent) i sends each of its neighbors j a binary-valued table $D_{j,i}$ indexed by all possible strategies of agents j and i. For any pair of strategies (s_j, s_i) a table $D_{j,i}$ is 1 if and only if there exists a BNE in which agent i plays s_i when its neighboring agent j plays s_j.

Consider a node i with neighbors j and $\mathcal{N}_{-i}^{j} = \{1, ..., m-1\}$. For induction, assume that each h sends node i table $D_{i,h}$. For any pair of strategies (s_j, s_i) a table $D_{j,i}$ is 1 if and only if there exists a vector of strategies $\overline{s}_{\mathcal{N}_{-i}^{j}} = \{s_1, ..., s_{m-1}\}$ (witness) for $\mathcal{N}_{-i}^{j}$ such that:
1. $D_{i,h}(s_i, s_h) = 1 \ \forall 1 \leq h \leq m$, and
2. s_i is the best-response to $\overline{s}_h$ and s_j.

There maybe more than one witness for $D_{j,i}(s_j, s_i) = 1$. In addition to computing the binary-valued tables (i.e. $D_{j,i}$), stage 1 of the algorithm also saves a list of witnesses for each pair of strategies (s_j, s_i) for which the table $(D_{j,i})$ is 1.

Now assume that $D_{j,i} = 1$ for some node i with neighbors j and $\mathcal{N}_{-i}^{j}$ for some witness $\overline{s}_{\mathcal{N}_{-i}^{j}}$. By construction, $D_{i,h}(s_i, s_h) = 1 \ \forall h$, and therefore by induction it must be that there exists BNE in which h plays s_h given that node i plays s_i and by construction of $D_{j,i}$ s_i is a best response of agent i and must be a part of BNE given that agent j plays s_j.

Stage 1 converges because all tables begin filled with 1 entries and entries can only change from 0 to 1,[79].

Stage 2 is a backtracking local assignment passing stage. It starts at an arbitrary node j which can chose any s_j for which $P(s_j) = 1$ and any witness $\overline{s}_{\mathcal{N}_{-i}^{j}}$ from the associated witness list. The node j then passes (s_j, s_h) to each its neighbors h telling h to play s_h.

```
Approximate-findBNE( )
Input: Game specification, ε approximation parameter, duration T
Output: ε-BNE of the game
Set t = 0; // initialize the prediction market's time period
Set r = 0; // initialize the round for the first stage
Set D^0_{j,i}(s_i, s_j) = 1 ∀s_i, s_j, ∀i, j; // initialize the table, where agent i and agent j are the
neighbors
foreach time period t ≤ T do
    Stage 1 : Local Optima
    while not converged do
        foreach agent pair i, j do
            foreach s_i, s_j do
                if D^r(s_j, s_h) = 1 ∀h AND s_j is Best Response to s̄_{N^j_{-i}} and s_i then
                    D^{r+1}(s_i, s_j) = 1 ;
                    save s̄_{N^j_{-i}} as a witness to D^{r+1}(s_i, s_j) = 1;
                end
                else
                    D^r(s_j, s_h) = 0 ∀h;
                end
            end
            send D^{r+1} to all of its neighbors;
        end
        r = r + 1 ;
    end
    Stage 2 : Global Optima
    foreach agent j do
        if ∃ s̄_{N^j_{-i}} s.t. D(s_j, s_h) = 1 ∀h ∈ N^j_{-i} then
            P_j(s_j) = 1;
        end
        else
            P_j(s_j) = 0;
        end
        // construct BNE - local search
        1. Pick any agent j and any s_j s.t. P(s_j) = 1 with witness s̄_{N^j_{-i}} ;
        2. Agent j assigns itself s_j and each of its neighbors h it assigns s_h;
        3. Pick next node and assign of all its unassigned neighbors with its witness;
        4. Backtrack if s_j is not the Best Response when all of the agent j's neighbors are
        assigned;
    end
    t++;
end
```

Algorithm 3: Algorithm to find BNE in a WBGG.

From the semantics of this message passing and backtracking step if s_j turns out to not be the best response when all of j's neighbors are assigned strategies, it must be true that s_j is the best response to its neighbors for any node j. $\square$

7.1.3 Computing approximate Bayes-Nash Equilibrium

Algorithm 3 is incompletely specified because the representation and computation of the step of passing tables in stage 1 is not completely specified. Since the strategy for an agent is a mapping from types to the simplex of probability distributions over actions, it may not be possible to represent tables D compactly or finitely for an arbitrary graphical game. We now present an algorithm for computing approximate BNE in incomplete information general structured graphical games with discrete types.

We adopt our abstract algorithm to compute ϵ-BNE in graphical games of an arbitrary structure with discrete types. Our updated algorithm takes parameter ϵ as input, that specifies how close of an approximation to BNE we want to get. The strategy space is discretized analogous to[64], such that any agent can only choose actions with probabilities that are multiples of τ, for some τ, instead of playing an arbitrary mixed strategy in $[0, 1]$. For a graphical games that contains l actions the probability that each action will be selected is a multiple of τ with the sum of all probabilities being 1. Then any agent i will have $O(\frac{1}{\tau^{2(l-1)}})$ different strategies.

Approximate-findBNE( )
Input: Game specification and ϵ approximation parameter Output: ϵ-BNE of the game
Run Algorithm 3 with two changes:
1. Only consider type-conditional discretized strategies
2. Change the requirement of best response to ϵ-best response.

Algorithm 4: Approximation algorithm to find BNE in a weighted graphical game.

Theorem 4. *For any $\epsilon > 0$, $k = max_i \eta_i \ll n$, and the discretization parameter $\tau \leq \frac{\epsilon}{l^k(4klog(k))}$ Algorithm 4 computes ϵ-BNE for an arbitrarily structured graphical game with incomplete information.*

Proof. In[100] it was shown that if the mixed strategy space for every type is restricted to multiples of τ, then for any ϵ and $\tau \leq \frac{\epsilon}{l^k(4klog(k))}$ there exists ϵ-BNE in tree structured graphical games. Their result however does not depend on the underlying graph being a tree, and therefore holds for arbitrary graphs also. The witness lists and tables of Algorithm 4 represent all ϵ-BNE. Therefore, Algorithm 4 is guaranteed to converge to an ϵ-BNE. □

Theorem 5. *For arbitrary structured graphical games with discrete types stage 1 in Algorithm 4 converges in at most $\frac{nk}{\tau^{4(l-1)}}$ rounds.*

Proof. The total number of entries in each table D is $O(\frac{1}{\tau^{2*2(l-1)}})$ since the number of entries is determined by the number of joint strategies of two agents with two possible types. Every round r before the algorithm converges has to change at least one entry in one table. Therefore, stage 1 of the Algorithm 4 has to converge in at most $\frac{nk}{\tau^{4(l-1)}}$ rounds, where k is the maximum degree of any node in the graph. □

Since our work in this chapter extends the existing NashProp algorithm, we don't expect scalability and complexity to be significantly different from[79]. Instead we report the dynamics in market maker's prices and utilities which demonstrate the behavior, important features and successful operation of our model in a distributed prediction markets setting.

Proposition 4. *Algorithm 4 applied to a distributed prediction market problem that uses utility function given in Equation 7.2 encourages truthful revelation.*

Proof. For the simplicity of notation we show the proof for two agents i and j, with two possible types θ_{iopt} and θ_{ipes}, but the proof is extendable to multiple agents with several possible types. We want to show that the expected utility that the agent i gets when choosing action A_i^{true} truthfully is greater or equal to the expected utility it gets when it chooses action A_i^{false}, i.e. $EU_i^{true} \geq EU_i^{false}$. Since there are only two possible actions $A_i = \{\lambda_i, -\lambda_i\}$, misreporting would mean that when Algorithm 1 recommends agent i to take action $A_i = \lambda_i$, it takes action $-\lambda_i$ instead; i.e. $A_i^{false} = -A_i^{true}$. From definition of

Name	Symbol	Value
Number of market makers	N	20
Market price's jump scaling parm.	ζ	0.01
Confidence that other market maker is of the same type	α	0.5
Prob. distr. over types	p	$U[0,1]$
Discretization parameter	τ	0.3
Approximation value	ϵ	0.05
Number of actions	$l_i \forall i$	2
Number of neighbors	$\eta_i \forall i$	8

Table 7.2: Parameters used for our simulation experiments.

utility given in Equation 7.2 we can express the utility as $u_i(\lambda_i) = (T-t)e^{-|A_i|(T-t)}$. Now,

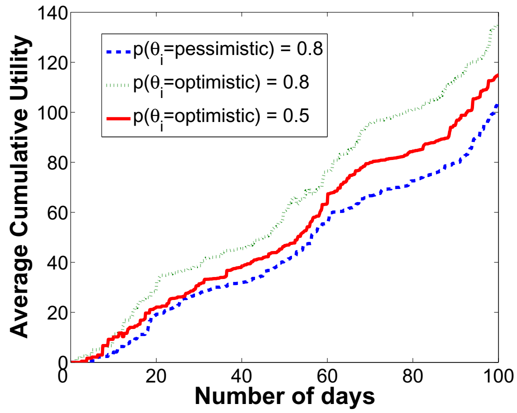
$$
\begin{aligned}
&EU_i^{true} = \alpha p(\theta_{jopt}|\theta_{iopt})(T-t)\left(e^{-|A_i^{true}|(T-t)}\right)\\
&+(1-\alpha)p(\theta_{jpes}|\theta_{iopt})(T-t)\left(e^{-|A_i^{true}|(T-t)}\right)\\
&\text{and } EU_i^{false} = \alpha p(\theta_{jopt}|\theta_{iopt})(T-t)\left(e^{-|-A_i^{true}|(T-t)}\right)\\
&+(1-\alpha)p(\theta_{jpes}|\theta_{iopt})(T-t)\left(e^{-|-A_i^{true}|(T-t)}\right).
\end{aligned}
$$

Since $|A_i^{true}| = |-A_i^{true}|$, we get $EU_i^{true} = EU_i^{false}$. Therefore, agent i does not have any incentive to reveal its action untruthfully. □

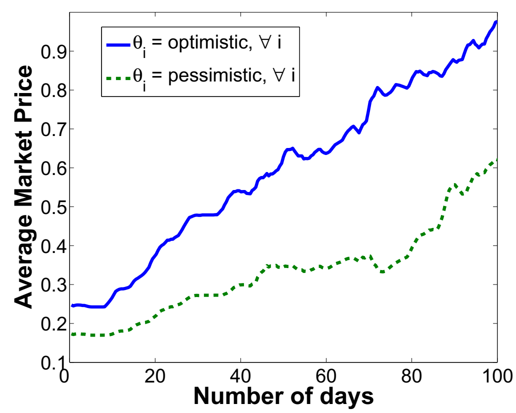
7.2 Experimental Results

We conduct several simulations using our algorithm for a distributed prediction market setting to observe and verify the effect of different parameters on the evolution of market maker utilities and prices in the markets. To make it easier to analyze the effect of different parameters, in all of our simulations we assume that the number of neighbors is fixed for each market maker. For each experiment we vary one set of parameters as specified in each set of experiments and we hold the other parameters fixed at their default values given in Table 7.2.

For our first set of experiments, we vary p, the probability distribution over types.



a



b

Figure 7.2: The average cumulative utility(a) and the average market price(b) for different types of market makers.

We allow for three types of market maker populations: mostly pessimistic, equal number of optimistic and pessimistic, and optimistic. Figure 7.2(a) shows the cumulative utility averaged over all 20 market makers, where the market maker population is either 80% pessimistic, 50% pessimistic and 50% optimistic, and 80% optimistic. The type of each market maker is determined at the beginning of the prediction market and it does not change over time. We observe that when the majority of market makers is pessimistic the average utility is 23% less than when the majority of market makers is optimistic. This is because optimistic market makers' strategy selection is affected by the average strategies of their neighbors, whereas pessimistic market makers' strategy selection is affected by just one other market maker choosing a pessimistic strategy. In Figure 7.2(b) we continue analyzing the effect of different market makers' types by looking at the average market price produced by the optimistic and the pessimistic market makers for the outcome of the event that happens (market price = 1). We note that optimistic market makers are able to predict a more accurate market price as opposed to the pessimistic ones. Again, this is due to the optimistic market makers taking into account the average strategies of all the market makers in their neighborhood. However, in real prediction market, there may be a mix of different types of market makers. Therefore, for the default setting for the type distribution we assume that there is about the same number of optimistic market makers as pessimistic ones.

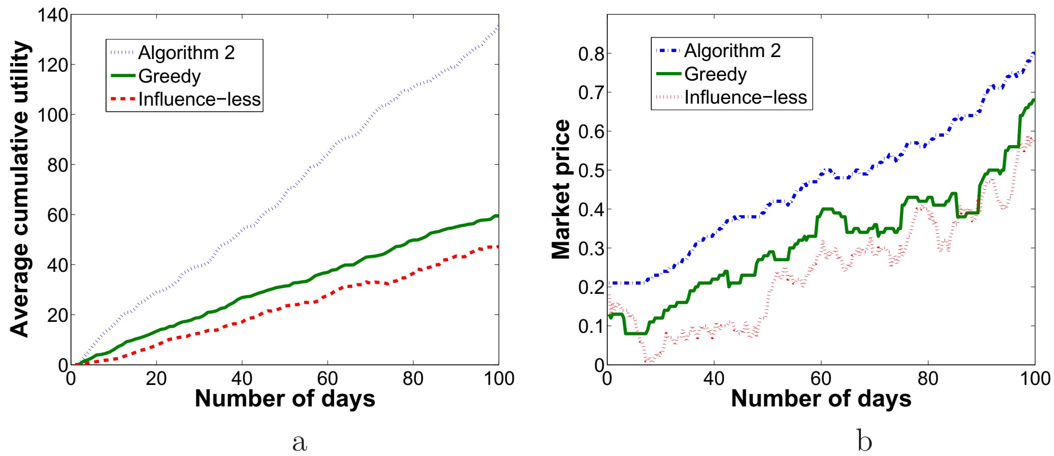


Figure 7.3: The average cumulative utility(a) and the market price(b) of one market maker using Algorithm 4, Greedy, or Influence-less markets strategies.

Next, we compare our Algorithm 4 to two other strategies:

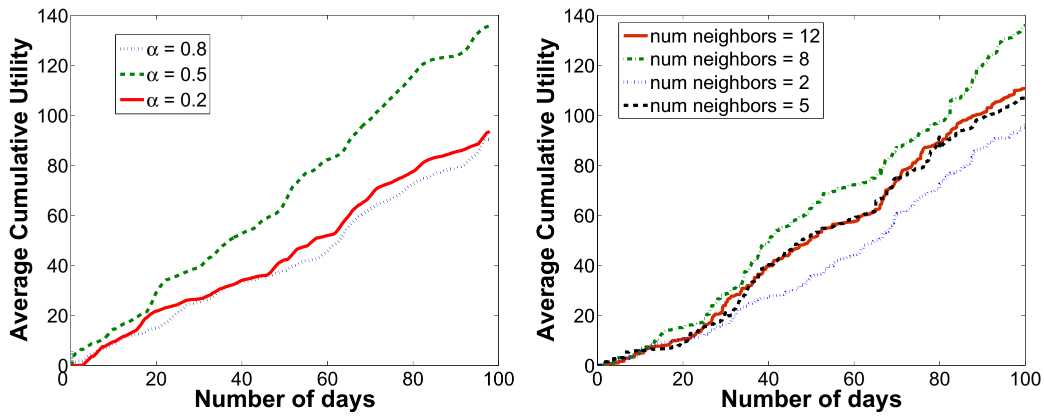
- **Greedy strategy:** In this setting, each agent i chooses λ_i that maximizes immediate utility given in Equation 7.2. This strategy does not consider the types of the market makers.

- **Influence-less markets**: In this setting, we consider conventional single, isolated markets where the market price is determined by the market maker based on that market's traders' decisions only. This setting is the completely opposite scenario of a distributed prediction market and it captures the effect of inter-market influences on the market makers' utilities and prices. To abstract the details of the traders' decisions, we have assumed that each agent i uses a derivative follower (DF)[40] strategy where it keeps on increasing its market price π_i by δ_i until its immediate utility u_i starts decreasing, at that time agent i starts decreasing π_i by δ_i. This strategy does not consider the types or the interaction among market makers.

Figures 7.3 (a) and (b) show the utilities and market prices for the market makers using our proposed algorithm, greedy strategy, or influence-less markets strategy correspondingly. We observe that market makers using our Algorithm 4 obtain 56% more utility than the market makers following the next best greedy strategy. We also note that market prices fluctuate more when market makers use a greedy strategy or are in influence-less markets than when they use WBGG because greedy and influence-less market strategies are myopic and do not consider market makers' types. Finally, this result provides an important justification of our work in this part of the thesis - that, as compared to influence-less markets with isolated market makers, interacting market makers in a distributed prediction market are able to improve their utilities and predict prices with less fluctuations.

For our next set of experiments we analyze the effect of the α parameter which is used in Equation 7.1. This parameter is the confidence parameter representing market maker agent i's belief that the other market maker in its neighborhood is of the same type as itself. We assume that α is the same for all of market maker agent i's neighbors and it is set at the beginning of the market. We can see from Figure 7.4(a) that for the setting where the market makers are of mixed types (about half is pessimistic and the other half is optimistic), the market makers get the highest utility when $\alpha = 0.5$. However, when market makers are confident that all of their neighbors are either of the opposite type or of the same type, they get $32\% - 33\%$ less utility correspondingly.

Finally, we report the results for different number of market makers and different number of neighbors of each market maker. Figure 7.4(b) shows the cumulative utility averaged over neighboring market makers for a setting with 20 market makers. We can see that when market makers have a small number of neighbors then they get less utility than when the number of neighbors is larger, up to a certain point. For example, market makers with 2 neighbors (10% of the total number of market makers) get 29% less utility than when market makers have 8 neighbors (40% of the total number of market makers). However, this relationship is not linear, i.e. when market makers have 12 neighbors (60% of the total number of market makers) they get 19% less utility than when market makers have 8 neighbors. We posit that increasing the number of neighbors up to a point translates to an increased utility because the market maker can improve its decision based on the



a b

Figure 7.4: The average cumulative utility for different values of α(a), the cumulative utility averaged over all neighbors for different number of neighbors, i.e. $2, 5, 8$, and 12(b).

information of its neighbors, but having too many neighbors may end up creating more noisy information for the market maker.

We also conduct experiments showing the scalability of our Algorithm 4 with respect to the number of market makers, and our results show that our algorithm scales linearly with the number of market makers and that the running time increases with the increased number of neighbors.

In summary, in this chapter we proposed a novel distributed prediction markets setting where the aggregated (market) price of a security of an event in one prediction market is affected dynamically by the prices of securities of similar events in other, simultaneously running prediction markets. Our proposed formal framework, called a weighted Bayesian graphical game (WBGG), is able to capture the local interactions between multiple market makers and uses the Bayes-Nash equilibrium concept to find a suitable action for each market maker in WBGG. Our experimental results showed our algorithm results in higher utilities and more accurate prices in comparison to a greedy strategy or a disjoint prediction markets. The work presented in this part of the thesis was published in[59] and the overview of our findings is shown in Table 7.3.

Contr.	Research Question	Research Findings
5	How can we design a distributed model of a prediction market to be used for decision making?	- We use a model based on a graphical game that allows to model interactions of multiple prediction markets.
5	Is our distributed model of the prediction market incentive-compatible?	- We prove that our model encourages truthful revelation.
5	How does a distributed information aggregation in a prediction market compare to the centralized aggregation?	- Our experiments show that market makers get higher utilities and achieve better accuracy in a distributed setting.

Table 7.3: Research findings for research questions in contribution 5 of this thesis.

Chapter 8

Conclusions

This thesis makes significant contributions to an analysis of prediction markets using a multi-agent system, game theory, and Boolean network-based tools and to using the generated knowledge for the design of information aggregation systems that can be used for decision making and in a distributed scenarios. In this chapter we summarize our contributions and identify future directions for our research. We also identify general open problems in prediction markets.

8.1 Lessons Learned and Future Directions

This thesis makes five distinct contributions, which are summarized below along with future research directions for each contribution:

1. The first contribution is a multi-agent system that is used to analyze the effect of information on the prediction market performance. It helps answering the question of how the changes in different aspects of information affect the performance of prediction markets. Our research showed that incorporating information related parameters into prediction market models can give insights into realistic aspects of prediction market's performance. For example, we verified that when information arrives more frequently prices fluctuate less and utilities increase, when information is more reliable prices are more stable and utilities are higher, and when information

is easily available prices are driven down and utilities are higher. A direction worth future investigation is how to incorporate psychological and sociological data about humans' beliefs and their private information. A possible way to address this issue is to run a controlled real prediction market experiment to obtain human trader's data, and then use this data to further test the effect of information on prediction markets. We are also interested in introducing and investigating correlations between events in a prediction market.

2. The second contribution is a correlated equilibrium strategy for the traders within a partially observable stochastic game-based model that incentivizes traders to reveal their true beliefs and allows them to achieve higher payoffs than existing trading strategies. This study uses the real data to compare the performance of the multi-agent system based prediction market to the real prediction markets. From this contribution we learned that accounting for the uncertainty of the environment leads to a more precise prediction market model. We observed that game-theoretic correlated equilibrium strategy improves the accuracy of the market prices and the utility of the trading agents. Also, while using real data we were able to show that trading agents using correlated equilibrium trading strategy can avoid inefficient human trading decisions that might result in a very large loss. Our work uses the Logarithmic Market Scoring Rule (LMSR) to calculate the market price. However, LMSR has been recently shown to have some drawbacks, for example, the market maker can run at a loss which can be large and a single parameter b controls the loss bound, the level of liquidity, and the rate of adaptability to market shocks[83]. Also, in our work, we used the correlated equilibrium to determine the traders' or agents' selected strategies. Although the CE finds a stable strategy for each agent, that strategy may not satisfy certain desirable properties such as maximizing the joint utility received by the agents. Besides, the CE strategy also relies on a third party, the market-maker in the prediction market, to calculate the probabilities with which the agents could play different strategies. An alternate solution concept to the CE that could guarantee desirable properties such as maximizing joint utilities to agents, guaranteeing

a minimum utility to each agent, etc., and reducing or removing the reliance on the market maker to calculate the CE would be worth future investigation. Another future direction worthy of investigation is to incorporate liquidity-sensitive market-maker that has been proposed very recently[87] for aggregating the traders' or agents' information. We also plan to study a technique for choosing the best equilibrium efficiently, as it currently lies outside of the scope of this project.

3. The third contribution is a study of the dynamics of prediction markets under various conditions using Boolean Network techniques. The results of this study show how various market parameters affect the prediction market's behavior. An interesting lesson that we learned from this study is that using simple boolean rules to model belief update of the traders does not negatively affect the quality of the aggregated market price. We also observed that the market price can stabilize with the presence of noise and also with increasing number of traders. In this work we assumed the simple synchronous information transmission since our goal was to study the overall dynamics of the market. Future directions to extensively study this aspect include incorporating asynchronous information transmission that can be applied in a deterministic or stochastic way and studying different belief update strategies based on game theory.

4. The fourth contribution is a system that uses prediction markets for decision making. This system is applied to sensor fusion domain and is tested in the landmine detection scenario. In this study we showed that a special payment function can be used to incentivize the sensor agents to report their observations truthfully. We also observed that having different types of sensors participate in a simulated prediction market can lead to better decision making and improve object type identification accuracy. Future directions to extend our model could be to implement the proposed techniques on physical robotic and sensor systems, minimizing the detection time while guaranteeing the most accurate detection, integrating the decision making problem and scheduling problem facing the decision maker, and achieving budget balance in our proposed

mechanism.

5. The fifth contribution is a distributed-computational model of a decentralized information aggregation between multiple prediction markets that are able to interact and affect each other's aggregated market prices. This study not only models important and realistic distributed prediction markets scenario, but also provides an algorithm to be used by agents in the distributed prediction market scenario that outperforms the greedy strategy and the agents in the setting where prediction markets are disjoint. In this contribution we found that prediction markets that are running similar events can affect each other's aggregated market prices. We learned that the game-theoretic Bayes-Nash based algorithm improves the accuracy of the prediction markets and also provides higher utilities to the traders in such inter-markets in comparison with disjoint prediction markets. While our model studies the interaction among multiple market makers, it implicitly incorporates influences between traders in multiple prediction markets through the market maker's prices. In the future, we plan to extend our framework to incorporate traders across different prediction markets. We also plan to extend our algorithm for distributed prediction markets setting to study stochastic graphical games which can model uncertainty as well as some repeated games.

8.2 Open Problems

We finally conclude with some more general open problems that are worthy of investigation to get a better understanding of prediction markets. Market scoring rules and automated market makers in prediction markets have specifically generated a lot of research in the past few years. However, there are still many computational challenges in using automated market makers, especially in combinatorial prediction markets. There has also been a lot of recent progress made in the direction of analyzing agents' behaviors in prediction markets, with one of the biggest research concentrations being on incentive compatibility and manipulation. However, the full impact of these aspects can be understood with more work on non-myopic and non-risk-neutral agent behavior. There are a lot of open and interesting problems to be solved in using prediction markets for decision making, especially in generating incentive-compatible decision markets with many agents. Below we present a few specific open research questions related to prediction markets.

Open Questions:

- **Traders:**

  - What is the minimum number of traders that can guarantee a reasonably accurate prediction?

  - How does group trading affect the accuracy of the aggregated market price?

  - What is a suitable strategy for a trading agent to choose its bets in a combinatorial prediction market that are perhaps subject to constraints or penalties on the number or complexity of bids?

- **Prediction Market:**

  - What is a suitable model, e.g. filtering, for aggregating traders' prices under noisy conditions?

  - What are the constraints on accurate predictions (type of market, type of event, type of trader, etc.) in a prediction market?

  - How can the aggregation computation be done in a distributed manner without using a centralized entity like the market maker?

We envisage that with more research and a better understanding of prediction markets, they can be used as an effective paradigm to address issues and challenges in several applications of distributed intelligence. With the recent prominence of large scale data and large scale networks, prediction markets can be used as an important tool in analyzing "big data" and in harnessing human knowledge for the purpose of making better decisions. Prediction markets can also be combined with social networks or sensor networks to create a large volume market that can be used to aggregate the large volume of data effectively. Finally, the frameworks and algorithms that we developed in this thesis can be extended beyond prediction markets. We foresee that our methods can also be applied to social media and complex, large-scale networks.

References

[1] Agrawal, S., Delage, E., Peters, M., Wang, Z. & Ye, Y. (2011). A unified framework for dynamic prediction market design. *Operations Research*, **59**, 550–568. 18

[2] Aumann, R. (1974). Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, **1**, 67 – 96. 60

[3] Avery, C. & Zemsky, P. (1998). Multidimensional uncertainty and herd behavior in financial markets. 724–748. 119

[4] Barbu, A. & Lay, N. (2012). An introduction to artificial prediction markets. *Journal of Machine Learning Research (forthcoming)*. 34

[5] Berg, J. & Rietz, T. (2003). Prediction markets as decision support systems. *Information Systems Frontiers*, **5**, 79–93. 29

[6] Betfair (2012). Betfair. `http://www.betfair.com/`. 12

[7] Bilke, S. & Sjunnesson, F. (2001). Stability of the kauffman model. *Physical Review*, **65**. 90

[8] Bleichrodt, H., van Rijn, J. & Johannesson, M. (1999). Probability weighting and utility curvature in qaly-based decision making. *Journal of Mathematical Psychology*, **43**, 238–260. 59

[9] Boutilier, C. (2012). Eliciting forecasts from self-interested experts: Scoring rules for decision makers. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '12, 737–744. 31

[10] Brahma, A., Chakraborty, M., Das, S., Lavoie, A. & Magdon-Ismail, M. (2012). A bayesian market maker. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, EC '12, 215–232. 20

[11] Brier, G. (1950). Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, **78**, 1–3. 16

[12] Chen, Y. & Kash, I. (2011). Information elicitation for decision making. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '11, 175–182. 30, 31, 103, 105

[13] Chen, Y. & Pennock, D. (2007). Utility framework for bounded-loss market maker. In *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence (UAI '07)*, 49–56. 6, 18, 21, 27, 79, 88

[14] Chen, Y. & Vaughan, J. (2010). A new understanding of prediction markets via no-regret learning. In *Proceedings of the 11th ACM conference on Electronic commerce*, EC '10, 189–198. 18, 21

[15] Chen, Y., Chu, C.H., Mullen, T. & Pennock, D. (2005). Information markets vs. opinion pools: an empirical comparison. In *Proceedings of the 6th ACM conference on Electronic commerce*, EC '05, 58–67. 14

[16] Chen, Y., Mullen, T. & Chu, C. (2006). An in-depth analysis of information markets with aggregate uncertainty. *Electronic Commerce Research*, **6**, 201–222. 2, 24

[17] Chen, Y., Fortnow, L., Lambert, N., Pennock, D. & Wortman, J. (2008). Complexity of combinatorial market makers. In *Proceedings of the 9th ACM conference on Electronic commerce*, EC '08, 190–199. 19

[18] Chen, Y., Dimitrov, S., Sami, R., Reeves, D., Pennock, D., Hanson, R., Fortnow, L. & Gonen, R. (2009). Gaming prediction markets: Equilibrium strategies with a market maker. *Algorithmica*, **58**, 930–969. 6, 26, 27, 104

[19] Chen, Y., Kash, I., Ruberry, M. & Shnayder, V. (2011). Decision markets with good incentives. In *The 7th Workshop on Internet and Network Economics*, 72–83. 31

[20] Chen, Y., Xi, A., Goldstein, R. & Kash, I. (2011). Market manipulation with outside incentives. In *Proceedings of the Twenty-fifth AAAI Conference on Artificial Intelligence*, AAAI-11, 614–619. 32

[21] Chen Y., M.T. & Chu, C.H. (2004). Theoretical investigation of prediction markets with aggregate uncertainty. In *Proceedings of the 7th International Conference on Electronic Commerce Research (ICECR)*, 81–90. 77

[22] Conitzer, V. (2009). Prediction markets, mechanism design, and cooperative game theory. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI '09)*, 101–108. 27

[23] Cowgill, B., Wolfers, J. & Zitzewitz, E. (2008). Using prediction markets to track information flows: Evidence from google. Tech. rep., Dartmouth College, bocowgill.com/GooglePredictionMarketPaper.pdf. 4, 24

[24] Cremer, F., Schutte, K., Schavemaker, J. & den Breejen, E. (2001). A comparison of decision-level sensor-fusion methods for anti-personnel landmine detection. *Information Fusion*, **2**, 187–208. 100, 101, 108

[25] Dempster, A., Laird, N. & Rubin, D. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of Royal Statist. Soc. Ser. B*, **39**, 1–38. 68

[26] Dimitrov, S. (2010). *Information Procurement and Delivery: Robustness in Prediction Markets and Network Routing*. PhD Thesis, University of Michigan. 58

[27] Dimitrov, S. & Sami, R. (2010). Composition of markets with conflicting incentives. In *Proceedings of the 11th ACM conference on Electronic commerce*, EC '10, 53–62. 32

[28] Dimitrov, S., Sami, R. & Epelman, M. (2009). Subsidized prediction markets for risk averse traders. In *Proceedings of the 5th International Workshop on Internet and Network Economics*, WINE '09, 491–497. 28

[29] Dudik, M., Lahaie, S. & Pennock, D. (2012). A tractable combinatorial market maker using constraint generation. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, EC '12, 459–476. 21

[30] Ellis, C. & Sami, R. (2012). Learning political science with prediction markets: An experimental study. *Political Science and Politics (forthcoming)*. 33

[31] Farfel, J. & Conitzer, V. (2009). Turing trade: A hybrid of a turing test and a prediction market. In *Auctions, Market Mechanisms and Their Applications, First International ICST Conference*, AMMA'09, 61–73. 32

[32] Feigenbaum, J., Fortnow, L., Pennock, D. & Sami, R. (2005). Computation in a distributed information market. *Theoretical Computer Science*, **343**, 114–132. 23

[33] Fudenberg, D. & Tirole, J. (1991). *Game Theory*. MIT Press. 55

[34] Gjerstad, S. (2005). Risks, aversions and beliefs in predictions markets. mimeo, University of Arizona. 13, 37, 38, 42, 45, 58, 84

[35] Glosten, L. & Milgrom, P. (1985). Bid, ask and transaction prices in a specialist market with heterogeneously informed traders. *Journal of Financial Economics*, **14**, 71–100. 20

[36] Goel, S., Reeves, D., Watts, D. & Pennock, D. (2010). Prediction without markets. In *Proceedings of the 11th ACM conference on Electronic commerce*, EC '10, 357–366. 14

[37] Goeree, J., Holt, C. & Palfrey, T. (2003). Risk averse behavior in generalized matching pennies games. *Games and Economic Behavior*, **45**, 97–113. 65

[38] Golub, B. & Jackson, M. (2010). Naive learning in social networks and the wisdom of crowds. *American Economic Journal*, **2**, 112–149. 79

[39] Goodrich, C. & Matache, M. (2007). The stabilizing effect of noise on the dynamics of a boolean network. *Physica A*, **379**, 334–356. 90

[40] Greenwald, A. & Kephart, J. (1999). Shopbots and pricebots. In *Sixteenth International Joint Conference on Artificial Intelligence, Stockholm*, 506–511. 134

[41] Gros, B. & Bruschini, C. (1996). Sensor technologies for the detection of antipersonnel mines: a survey of current research and system developments. In *Proceedings of the International Symposium on Measurement and Control in Robotics*, 509–518. 97

[42] Gruca, T., Berg, J. & Cipriano, M. (2008). Incentive and accuracy issues in movie prediction markets. *The Journal of Prediction Markets*, **2**, 29–43. 4

[43] Guo, M. & Pennock, D. (2009). Combinatorial prediction markets for event hierarchies. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '09, 201–208. 18, 19

[44] Hall, M., Frank, H.G., E., Pfahringer, B., Reutemann, P. & Witten, I. (2009). The weka data mining software: An update. *ACM SIGKDD Explorations Newsletter*, **11**, 10–18. 68

[45] Hansen, E., Bernstein, D. & Zilberstein, S. (2004). Dynamic programming for partially observable stochastic games. In *Proceedings of the 19th National Conference on Artificial Intelligence*, 709–715. 53, 55, 72

[46] Hanson, R. (2003). Combinatorial information market design. *Information Systems Frontiers*, **5**, 105–119. 12, 16, 17, 26, 27, 29

[47] Hanson, R. (2007). Logarithmic market scoring rules for modular combinatorial information aggregation. *Journal of Prediction Markets*, **1**, 3–15. 17, 27, 104

[48] Hanson, R. (2009). A manipulator can aid prediction market accuracy. *Economica*, **76**, 304–314. 91

[49] Holt, C. & Laury, S. (2002). Risk aversion and incentive effects. *American Economic Review*, **92**, 1644–1655. 59

[50] HSE (2012). Hollywood stock exchange. `http://www.hsx.com/`. 12

[51] Intrade (2012). Intrade. `http://www.intrade.com/`. 6, 13, 65

[52] Iyer, K., Johari, R. & Moallemi, C. (2010). Information aggregation in smooth markets. In *Proceedings of the 11th ACM conference on Electronic commerce*, EC '10, 199–206. 28

[53] Jian, L. & Sami, R. (2012). Aggregation and manipulation in prediction markets: Effects of trading mechanism and information distribution. *Management Science*, **58**, 123–140. 26, 27, 79

[54] Jumadinova, J. & Dasgupta, P. (2011). A multi-agent prediction market based on partially observable stochastic game. In *Proceedings of the* 13*th International Conference on Electronic Commerce*, ICEC '11, 21:1–21:10. 74

[55] Jumadinova, J. & Dasgupta, P. (2011). A multi-agent system for analyzing the effect of information on prediction markets. *International Journal of Intelligent Systems*, **26**, 383–409. 36, 51, 55

[56] Jumadinova, J. & Dasgupta, P. (2011). Partially observable stochastic game-based multi-agent prediction markets. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '11), (short paper)*, 1217–1218. 74

[57] Jumadinova, J. & Dasgupta, P. (2012). Multi-sensor information processing using prediction market-based belief aggregation. In *Proceedings of the 14th International Workshop on Agent-Mediated E-Commerce (AMEC-XIV)*. 117

[58] Jumadinova, J. & Dasgupta, P. (2012). Multirobot autonomous landmine detection using distributed multisensor information aggregation. In *Proceedings of the SPIE Conference on Defense Security and Sensing, Baltimore, MD*, SPIE 8407, Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications. 117

[59] Jumadinova, J. & Dasgupta, P. (2013). Distributed prediction markets modeled by weighted bayesian graphical games. In *Proceedings of the 15th International Workshop on Agent-Mediated E-Commerce (AMEC-XV)*. 136

[60] Jumadinova, J., Matache, M. & Dasgupta, P. (2011). A multi-agent prediction market based on boolean network evolution. In *Proceedings of the IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, WI-IAT '11, 171–179. 93

[61] Jurgen, D. (1993). Price competition between market makers. *The Review of Economic Studies*, **60**, 735–751. 121

[62] Kadane, J. & Winkler, R. (1988). Separating probability elicitation from utilities. *Journal of the American Statistical Association*, **83**, 357–363. 58

[63] Kearns, M. (2007). *Graphical games. In N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani (Eds.), Algorithmic Game Theory*. Cambridge, U.K.: Cambridge University Press. 124

[64] Kearns, M., Littman, M. & Singh, S. (2001). Graphical models for game theory. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, 253–260. 119, 123, 130

[65] Kollock, P. (1999). The production of trust in online markets. *Advances in Group Processes*, **16**, 99–123. 39

[66] Luce, R. & Krumhansi, C. (1988). Measurement, scaling, and psychophysics. *In Stevens Handbook of Experimental Psychology, Wiley: New York*, **1**, 3–74. 59

[67] Ma, H. & Leung, H. (2008). *Bidding Strategies in Agent-Based Continuous Double Auctions*. Whitestein Series In Software Agent Technologies And Autonomic Computing. 71

[68] Makarenko, A. & Durrant-Whyte, H. (2006). Decentralized bayesian algorithms for active sensor networks. In *Information Fusion 7*, 418–433. 96

[69] Manski, C. (2006). Interpreting the predictions of prediction markets. *Economic Letters*, **91**, 425–429. 13, 37, 58

[70] Manyika, J. & Durrant-Whyte, H. (1995). *Data fusion and sensor management*. Prentice Hall. 96, 115

[71] Marketplace, I.E. (2010). Iowa electronic marketplace. http://www.biz.uiowa.edu/iem. 46

[72] Michiardi, P. & Molva, R. (2002). Simulation-based analysis of security exposures in mobile ad hoc networks. In *Mobile Ad Hoc Networks. European Wireless Conference*. 96

[73] Milisavljeviċ, N. & Bloch, I. (2003). Sensor fusion in anti-personnel mine detection using a two-level belief function model, part c: Applications and reviews. *IEEE Transactions on Systems, Man, and Cybernetics*, **33**, 269–283. 96, 100, 101, 108, 112, 114

[74] Milisavljeviċ, N., Bloch, I. & Acheroy, M. (2000). Characterization of mine detection sensors in terms of belief functions and their fusion, first results. In *Proceedings of the Third International Conference on Information Fusion*, 15–22. 100, 101, 108, 109, 112

[75] Molm, L., Schaefer, D. & Collett, J. (2009). *Fragile and Resilient Trust: Risk and Uncertainty in Negotiated and Reciprocal Exchange*. Sociological Theory. 39

[76] Myerson, B. (1985). Bayesian equilibrium and incentive-compatibility: An introduction. In *Social Goals and Social Organization: Essays in Memory of Elisha Pazner*, 229–259. 125

[77] News, G. (2012). Google news. `http://news.google.com`. 68

[78] Nikolova, E. & Sami, R. (2007). A strategic model for information markets. In *Proceedings of the 8th ACM conference on Electronic commerce*, EC '07, 316–325. 18, 23

[79] Ortiz, L. & Kearns, M. (2003). Nash propagation for loopy graphical games. In *In Neural Information Processing Systems*, 793–800. 119, 123, 124, 127, 128, 131

[80] Osborne, M., Roberts, S., Rogers, A., Ramchurn, S. & Jennings, N. (2008). Towards realt-time information processing of sensor network data using computationally efficient multi-output gaussian processes. In *IPSN*, 109–120. 96, 115

[81] Ostrovsky, M. (2009). Information aggregation in dynamic markets with strategic traders. In *Proceedings of the 10th ACM conference on Electronic commerce (EC'09)*, 253–254. 6

[82] Othman, A. (2008). Zero intelligence agents in prediction markets. In *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems*, 879–886. 37

[83] Othman, A. & Sandholm, T. (2010). Automated market-making in the large: the gates hillman prediction market. In *Proceedings of the 11th ACM conference on Electronic commerce*, EC '10, 367–376. 4, 18, 20, 21, 139

[84] Othman, A. & Sandholm, T. (2010). Decision rules and decision markets. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '10, 625–632. 29, 30, 31

[85] Othman, A. & Sandholm, T. (2010). When do markets with simple agents fail? In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1*, AAMAS '10, 865–872. 25, 89

[86] Othman, A. & Sandholm, T. (2012). Profit-charging market makers with bounded loss, vanishing bid/ask spreads, and unlimited market depth. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, EC '12, 790–807. 21

[87] Othman, A., Sandholm, T., Pennock, D. & Reeves, D. (2010). A practical liquidity-sensitive automated market maker. In *Proceedings of the 11th ACM conference on Electronic commerce*, EC '10, 377–386. 18, 20, 140

[88] Pan, W., Altshuler, Y. & Pentland, A. (2012). Decoding social influence and the wisdom of the crowd in financial trading network. In *IEEE Social Computing / PASSAT*, 203–209. 119

[89] Papadimitriou, C. & Roughgarden, T. (2008). Computing correlated equilibria in multi-player games. *Journal of ACM*, **1**, 14:1–14:29. 60, 62, 63

[90] Papadimitriou, C. & Roughgarden, T. (2012). Comments on "computing correlated equilibria in multi-player games". http://stanford.edu/~tim/papers/comment.pdf. 63

[91] Pennock, D. (2004). A dynamic pari-mutuel market for hedging, wagering, and information aggregation. In *Proceedings of 5th ACM conference on Electronic commerce ACM Press*, 170–179. 22

[92] Pennock, D. & Sami, R. (2007). *Computational aspects of prediction markets, In Algorithmic Game Theory*. Cambridge University Press, n. Nisan, T. Roughgarden, E. Tardos, V. Vazirani, editors. 15

[93] Pfeiffer, T. & Almenberg, J. (2010). Prediction markets and their potential role in biomedical research - a review. *Biosystems*, **102**, 71–76. 33

[94] Plott, C. & Chen, K. (2002). Information aggregation mechanisms: Concept, design and implementation for a sales forecasting problem. Tech. rep., Social Science working paper no. 1131, California Institute of Technology. 3

[95] Rogers, A., David, E. & Jennings, N. (2004). Selfish sensors in wireless microsensor networks. In *Proceedings of Proceedings of the 4th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '04, 1322–1323. 96

[96] Rosencrantz, M., Gordon, G. & Thrun, S. (2003). Decentralized sensor fusion with distributed particle filters. In *IPSN*, 55–62. 96

[97] Russel, S. & Norvig, P. (2002). *Artificial Intelligence: A Modern Approach*. Prentice Hall. 84

[98] Servan-Schreiber, E., Wolfers, J., Pennock, D. & Galebach, B. (2004). Prediction markets: Does money matter? *Electronic Markets*, **14**, 243–251. 25

[99] Shi, P., Conitzer, V. & Guo, M. (2009). Prediction mechanisms that do not incentivize undesirable actions. In *Proceedings of the 5th International Workshop on Internet and Network Economics*, WINE '09, 89–100. 31

[100] Singh, S., Soni, V. & Wellman, M. (2004). Computing approximate bayes-nash equilibria in tree-games of incomplete information. In *EC '05*, 81–90. 123, 124, 131

[101] Stone, P., Kaminka, G. & Rosenschein, J. (2009). Leading a best-response teammate in an ad hoc teams. In *Proceedings of International Workshop on Agent-Mediated E-Commerce*, 132–146. 96

[102] Surowiecki, J. (2005). *The wisdom of crowds*. New York: Anchor Books, iSBN: 0-385-72170-6. 2

[103] Thelwall, M., Buckley, K., Paltoglou, G., Cai, D. & Kappas, A. (2010). Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, **61**, 2544–2558. 68

[104] Vickrey, D. (2002). Multi-agent algorithms for solving graphical games. In *Proc. of the National Conference on Artificial Intelligence*, 345–351. 124

[105] Vinyals, M., Rodriguez-Aguilar, J. & Cerquides, J. (2011). A survey of sensor networks from a multiagent perspective. *Computer Journal*, **54**, 455–470. 96

[106] Wakker, P. (2008). Explaining the characteristics of the power (crra) utility family. *Health Economics*, **17**, 1329–1344. 59

[107] Waltz, E. & Llinas, J. (1990). *Multisensor Datafusion*. Artech House. 95

[108] Winkler, R. (1969). Scoring rules and the evaluation of probability assessors. *Journal of the American Statistical Association*, **64**, 1073–1078. 16, 17

[109] Wolfers, J. & Zitzewitz, E. (2004). Prediction markets. *Journal of Economic Perspectives*, **18**, 107–126. 3, 10, 13, 15, 79

[110] Wolfers, J. & Zitzewitz, E. (2005). Manipulation in political stock markets. University of North Carolina. 89

[111] Wolfers, J. & Zitzewitz, E. (2006). Interpreting prediction markets as probabilities. Stanford Graduate School of Business. 13, 38, 42