The Relative Accuracy of LMSR and CDA Prediction Markets

The Relative Accuracy of LMSR and CDA Prediction Markets

A thesis presented by
Nicholas Beasley

to Applied Mathematics
in partial fulfillment of the honors requirements
for the degree of Bachelor of Arts
Harvard College
Cambridge, Massachusetts

March 15, 2020

Abstract

Prediction markets are a highly successful forecasting method, and they have outperformed other methods (polls, regressions, etc.) in a variety of settings. Most markets are run with one of two mechanisms: the continuous double-auction (CDA), which resembles a stock market; and the logarithmic market scoring rule (LMSR), which has a market maker and a cost function to determine price. While a good deal is known about various benefits and drawbacks to each mechanism, relatively little is known about how they compare in terms of accuracy. We use trader belief data from a set of CDA markets to simulate the corresponding LMSR markets on those same events, and find that the two mechanisms generally have statistically similar performance on average. This holds true for a variety of parameter settings for the LMSR. However, which mechanism is superior in a given market is heavily dependent on liquidity. Leveraging this fact, we propose a hybrid algorithm that combines the predictions of the two mechanisms in a liquidity-dependent way. This algorithm is able to obtain better average accuracy in many cases.

Acknowledgements

I would like to thank my advisor, Prof. Yiling Chen, for helping to guide me through the prediction markets literature as well as for helping me brainstorm ideas; I wouldn't have been able to find this topic without her assistance. I also greatly appreciate the time she took to answer my questions and to meet with me weekly to discuss my progress.

In addition, I would like to thank Pavel Atanasov (formerly of the Good Judgement Project team), who answered a lot of my questions about the dataset and methodology used in the GJP's paper.

Contents

1 Introduction

Prediction markets are an increasingly popular method of aggregating information to forecast events, such as elections, sports events, and even corporate decisions. Previous research has shown that prediction markets do at least as well as, if not significantly better than polls, expert predictions, and statistical methods. For instance, the Iowa Electronic Markets forecast election outcomes and have been shown to be more accurate than polls [1]. Corporate prediction market forecasts at Google and Ford have outperformed expert sales forecasts [5].

Most prediction markets concern trades on a security that realizes value 1 if an outcome occurs and value 0 if it does not occur, and are set up as a continuous double auction (CDA). Traders can submit orders to buy or to sell, and if a buy order and sell order are compatible, they match and trade at the price of the first order to be submitted. However, this raises several issues, such as possible illiquidity with a low number of traders. Not only could there be a wide bid-ask spread, but in this setting, any offer to buy or sell might serve as a signal of superior information, discouraging other parties from accepting [4].

An alternative method which has gained popularity is using an automated market maker that follows a market scoring rule (this is often a *logarithmic* market scoring rule; hence the name LMSR) [8]. By having a market maker that is always willing to buy or sell, trades can be executed whenever there is a trader willing to trade. Moreover, with a cost function equivalent to a proper "scoring rule", this market maker makes it optimal for risk-neutral myopic agents to report their true assessments of an event's probability. Given that the goal of prediction markets is to make accurate predictions, it seems appropriate to investigate whether one of these mechanisms has superior forecast accuracy to another.

Some previous work has attempted this comparison in the lab ([9], [11]). While this provides a good baseline from which to think about the comparative accuracy of CDA and LMSR markets, there are a few issues with generalizing the results. Ledyard et al. mainly focuses on combinatorial markets in which there are 8 binary variables and up to 256 possible outcomes [11]. While the authors only run one market for each variable in the double auction, other mechanisms are set up in a way that allows traders to express their beliefs about the correlation between different variables. This may have hampered the performance of the CDA, preventing the results of the paper from being an accurate assessment of CDA and LMSR markets in a non-combinatorial setting.

Another issue is that both papers only have three or six traders per market. Because there are so few traders and a large number of states, low liquidity necessarily hampers the performance of the CDA, and both groups of authors point out that traders focus on a small subset of contracts. Only comparing the two mechanisms in settings of low liquidity gives an inherent advantage to the LMSR, a mechanism that does not require high liquidity in order for trades to occur.

Finally, laboratory experiments cannot capture many of the complexities of real-life events on which prediction markets may be run. Even the challenging environments in both papers are

not comparable to forecasting the outcome of an election or a baseball game, for example. In Healy et al., the posterior beliefs traders should have after observing information are actually analytically computable using Bayes' rule [9]; one can imagine the difficulty of updating one's priors after reading the NY Times, the Wall Street Journal, and the past week's polling data.

Our study is motivated by the lack of an experimental comparison of the two mechanisms in a setting where they are on roughly equal ground (i.e. a non-combinatorial setting with many more traders than contracts). Our first idea was to find events for which both LMSR and CDA prediction markets had been run, and to compare the accuracy of predictions in both markets. However, several roadblocks arose with this approach. First, instances of markets with different mechanisms being run on the same event (ex: NFL games) have become rarer, as notable prediction markets like InTrade and TradeSports have shut down in recent years due to regulations. Other sites, such as Betfair, had data that was hard to access because it required physical presence outside the U.S. Furthermore, LMSR prediction markets are hard to find data for in general.

Even if we had managed to acquire this data, the comparison would be far from perfect. Different markets might differ in how informed their traders were, how long markets were open for, how liquid markets were, the maximum amount traders could risk, etc. In comparing a CDA market operated by one site and an LMSR market operated by another, we would be seeing the aggregated effect of these factors in addition to inherent differences in aggregation quality stemming from the mechanism. Thus, we sought a dataset which effectively allowed identical traders to participate in both mechanisms simultaneously.

While this may seem difficult outside of an experimental setting, a recent paper (Dana et al. 2019) provides the opportunity to do exactly this [6]. In the paper, the authors (part of the Good Judgement Project at Penn) run a series of CDA prediction markets on world political events. Sample questions include "Will Iran blockade the Strait of Hormuz before 1 January 2014?" and "Will Angela Merkel win the next election for Chancellor of Germany?" In each of these markets, traders not only have to submit buy and sell orders with quantities and prices, but they have to report their true beliefs about the probability of the event while doing so. For instance, a trader who offers to buy 10 shares of "Yes" at \$0.70 in the Angela Merkel market might indicate that they think Merkel actually has an 80% chance of winning the election. In [6], the authors use these belief reports to compare the accuracy of CDA markets and some of their belief aggregation algorithms (which involve transformations such as removing old information, extremizing, and weighting by prior accuracy).

However, we can view the LMSR market maker as just another belief aggregation algorithm—it operates similarly in that it takes all belief reports and aggregates them with the help of a cost function. By simulating an LMSR market where agents trade according to the belief reports they made in the CDA market, we can see whether the LMSR would have yielded more accurate predictions. When doing so, it is important to note that unlike the design of the CDA, the design of the LMSR gives the market operator the ability to set the value of some parameters. One important parameter is the liquidity parameter, often referred to as b in the literature. This parameter determines how easy it is to move the instantaneous price of a contract (i.e.

how many contracts you have to buy to move the price a certain amount). Furthermore, we must also choose the coefficient of relative risk aversion (ρ) for the agents trading in the LMSR. In comparing the two mechanisms, we experiment with the values of these parameters to see if they affect accuracy and if the results are robust to changes in parameters.

The main contribution of our paper is to provide a direct comparison of the quality/accuracy of information aggregation provided by the CDA and LMSR mechanisms, with minimal interference from confounding factors. This is the first large-scale comparison of accuracy done outside of the lab (that we know of) and is also the first attempt at a direct comparison in which the same traders are effectively participating in both markets. By performing this comparison for several values of parameters relevant to the LMSR (the liquidity parameter b and coefficient of relative risk-aversion ρ) and investigating factors (most notably liquidity) which correlate with higher relative accuracy from one mechanism, we also contribute a baseline understanding of the situations in which one mechanism could outperform the other.

Furthermore, we contribute an approach that can boost prediction accuracy and improve on the predictions from both models by taking a linear combination of the two predictions where the coefficient depends on liquidity. This approach is similar to the one taken by Dana et al., who report that using the simple mean of the predictions of their belief aggregation algorithm and the CDA price results in Brier scores that are (statistically) significantly better than predictions made just by using prices. We take this a step further; rather than taking the average of the LMSR and CDA predictions, we leverage the fact that we expect the LMSR to be more accurate (relative to the CDA) in less liquid markets. We find that training the value of the linear combination coefficient on a training set of markets and applying it to a test set can result in less error, although the significance is parameter-dependent. This hybrid approach could be useful to future market operators looking to improve on the performance of individual mechanisms.

The remainder of the paper proceeds as follows. Section 2 gives a more detailed summary of related work. Section 3 gives background on how prediction markets are used, and how the CDA and LMSR mechanisms work. Section 4 presents the model we use for the LMSR and describes our methodology and dataset in more detail. Section 5 presents the results of the accuracy comparison between the two mechanisms and presents the effect of liquidity on the comparative accuracy of both mechanisms. Section 6 proposes a boosted prediction algorithm and compares its accuracy to that of the CDA and LMSR. Finally, sections 7 and 8 summarize the main findings and conclude.

2 Related work

The prediction markets literature contains a great deal of work comparing prediction markets to other methods, ranging from expert forecasts to regressions. They generally agree that prediction markets outperform more traditional mechanisms, though they disagree about the degree to which this is the case. For instance, in [1], the authors find that from the 1988 to 2004 presidential elections, the Iowa Electronic Markets (which use the CDA mechanism) predicted the final vote share more accurately than 74% of polls. Furthermore, when looking at polls taken more than 100 days before the election, the IEM's predicted vote share (at the time of each poll) was significantly more accurate in each of the five elections.

At the other end of the spectrum, Goel et al. compare predictions made on NFL games from TradeSports (an online prediction market), an incentivized poll, and a basic statistical model only utilizing home-field advantage and the two teams' win-loss record [7]. They find that while TradeSports performed the best in predicting point differential, the poll was only 1% worse and the statistical model was only 3% worse in terms of RMSE, despite only incorporating two factors. The authors then compare predictions of opening weekend box office revenues made by Hollywood Stock Exchange (a prediction market also known as HSX) and a log-log regression using number of screens and search frequency as independent variables. The log-log regression is only 6% worse in terms of RMSE.

Studies have also been conducted comparing mechanisms in an experimental/lab setting, while extending the comparison to include different types of prediction markets (i.e. LMSR and CDA). The first major paper in this area was arguably Ledyard et al.'s work on combinatorial prediction markets, where the authors compared the effectiveness of CDA, call markets, opinion pools, and market scoring rules by looking at Kullback-Leibler distance [11]. They find that the market scoring rule is more accurate than the other mechanisms in a "training" environment with three binary variables (8 possible states) and that it is still more accurate (though roughly tied with opinion pools) in a "challenging" environment with eight binary variables. The authors find that the double auction market performed the worst in both scenarios.

Another key paper in this area is Healy et al. [9]. The authors compare CDA, iterated polling, parimutuel betting, and the LMSR in a simple setting with two states and a complex setting with eight states (similar to Ledyard et al.). Unlike in Ledyard et al., the frequency of predictions inconsistent with Bayes' rule and number of periods of no trading are also tabulated and considered alongside accuracy (measured by "distance" from the true posterior distribution) in determining which mechanism is best. They find that the LMSR has the most error in the simple setting but that it also has (along with opinion polls) the lowest error in the complex setting. Furthermore, the LMSR outperforms the CDA in the other metrics in the complex setting.

3 Background on Prediction Markets

The main goal of prediction markets is to aggregate information held by a disparate group of individuals into an accurate prediction. Traditionally, these predictions could be aggregated by polling or surveying "experts" and then combining the predictions using an algorithm. However, this raises several issues. For example, how do we select who to solicit information from? Our methods for determining who has information may be highly flawed, and we may also be limited in the number of people we can reach. Once we get reports, new issues arise: how do we know that the responses reflect truthful beliefs, and how can we accurately aggregate them?

Prediction markets address many of these issues. By setting up a market that anyone can trade on, we reach a wider audience of respondents without having to make judgement calls on who to survey. By financially rewarding people for their information (if a person knows the current price is too low or too high, they can trade accordingly and profit), we incentivize accurate and well-informed responses. Furthermore, markets provide money-weighted predictions, as traders who buy or sell more shares can move the market price more. Ideally, the willingness of a trader to risk more money would correlate with the amount and reliability of information they have, providing an intuitive way to aggregate beliefs.

3.1 CDA

One way to run a prediction market is to use a **continuous double auction**, which operates much like a stock market. Traders submit buy orders (bids) or sell orders (asks) requesting to purchase or sell a certain number of shares at a designated price. These orders are maintained on an order book, and stay there until the user cancels them or a trade occurs. A trade occurs when a bid is made at a price greater than or equal to the lowest ask price, or an ask is made at a price less than or equal to the highest bid. In this scenario, a trade occurs at the price of the order that was already in the book. Trades keep occurring while there are bid(s) and ask(s) whose prices overlap in this manner. If an ask is big enough to trade against multiple bids on the order book, it trades against the highest bid first (and bids trade against the lowest ask first).

Example 3.1. Let the current order book be:

Bid price	Bid quantity	Ask price	Ask quantity
50	100	55	200
		60	100

Currently, no trades occur because the highest bid price is lower than the lowest ask price. The trader willing to buy 100 shares at 50 each is not offering to pay enough for the traders whose orders are in the ask column to want to sell to him. Now assume trader A places an order to *buy* 250 shares at 60. This bid overlaps with both asks on the book, as the sellers for both orders would be happy to sell shares for 60.

The first transaction that occurs is that trader A buys the 200 shares at 55. The price is 55 because this sell order was on the book before the bid was made. After this, trader A still

wants to buy 50 more shares at 60, so he buys 50 of the shares offered at 60. Following this trade, there is no more overlap, and the order book looks like this:

Bid price	Bid quantity	Ask price	Ask quantity
50	100	60	50

□

While this is not exact, we can roughly interpret the prediction market price (as measured by the price of the last trade, or the midpoint of the bid-ask spread) as the "mean" belief held by traders. Wolfers and Zitzewitz [16] show that for a variety of utility functions and belief distributions, the equilibrium price in a prediction market is a good approximation of the mean belief, although the approximation is not as good when traders are risk-neutral or beliefs are widely dispersed.

Notice that this structure relies on two agents being willing to trade with each other. One major issue with CDA markets is that they can be very illiquid when they have few traders. In this scenario, the bid-ask spread may become very wide (i.e. the highest bid price will be far below the lowest ask price), discouraging trades and preventing the market from reaching an equilibrium price that is a useful prediction. Furthermore, even when a trade does occur, the price (represented by the last trade) moves wildly, resulting in noisy predictions. A well-known proposed reason for this is the *no-trade theorem* of Milgrom and Stokey, which suggests that because CDA markets are zero-sum games (the total sum of everyone's payoffs must be 0), rational risk-neutral agents won't trade, as willingness to trade is an indicator of having better information [4].

3.1.1 A Real-World Example

One example of a well-known market that uses this mechanism is PredictIt. The user interface is slightly different than the setup that we have described, suggesting that there are two types of securities you can buy ("Buy Yes" and "Buy No"), but "buying no" at a price of p cents is equivalent to making a sell order at a price of $100 - p$ cents. For example, for the Bernie Sanders "yes" contract, the lowest ask is 65 cents and the highest bid is 64 cents (100 minus the best offer on "buy no").

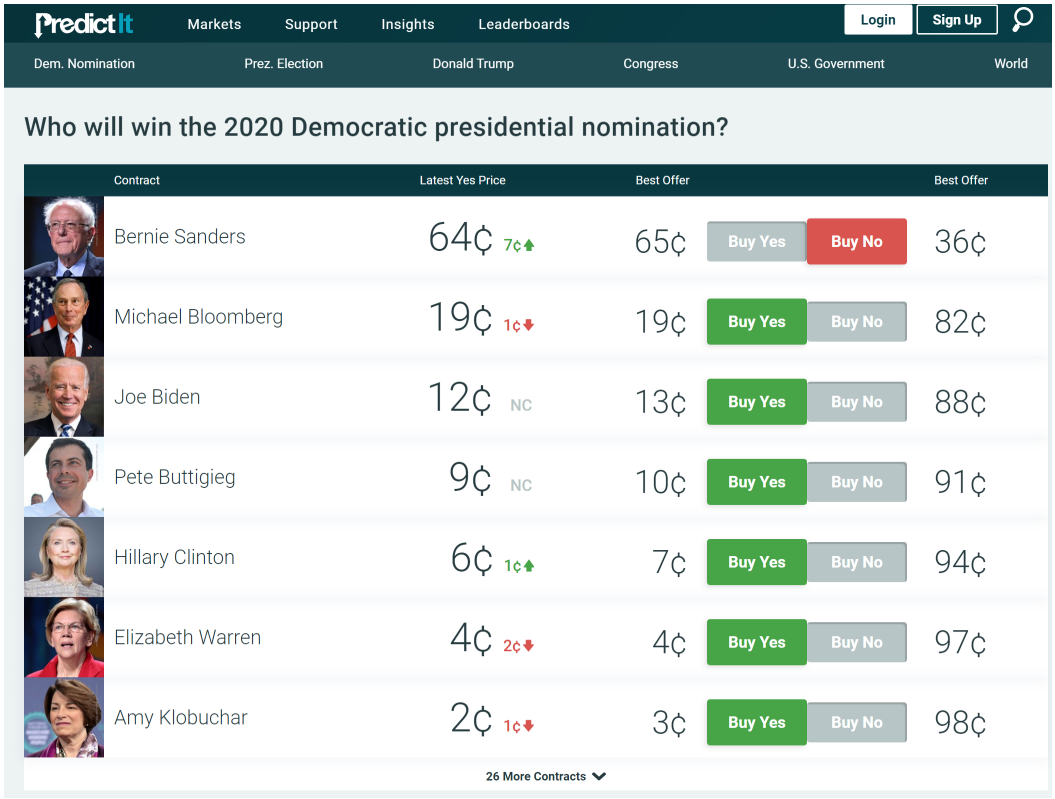


Figure 1: Screenshot of PredictIt, which runs CDA markets on elections and other political events

3.2 LMSR

An alternative mechanism for operating a prediction market is a **market scoring rule** (MSR), of which the *logarithmic* MSR is the most commonly used. In a market scoring rule, the mechanism starts with a prior distribution $\mathbf{q}_0$ (often uniform) over outcomes. Once the market opens, players can change the market's distribution to $\mathbf{q}_1, \mathbf{q}_2$, etc. If the current distribution of the market is $\mathbf{q}_j$ and the player changes it to $\mathbf{q}_{j+1}$, they ultimately receive a payment $s(\mathbf{q}_{j+1}) - s(\mathbf{q}_j)$, where s is a strictly proper scoring rule.

Definition 3.1. A **scoring rule** $s(q, o)$, where q is a reported probability distribution over the outcome space O and o is the outcome that occurs, is a function that specifies a real-valued reward for any $q \in \triangle O$ and $o \in O$.

Definition 3.2. Let a player's true beliefs over the outcome space be p_T. A **strictly proper** scoring rule is one for which:

$$\mathbf{E}_{o \sim p_T}[s(p_T, o)] > \mathbf{E}_{o \sim p_T}[s(q, o)] \quad \forall q \neq p_T.$$

In other words, your expected payoff from the scoring rule (based on your beliefs) is strictly maximized if you report your true beliefs as opposed to any other belief distribution.

This property is useful in general because using a strictly proper scoring rule should elicit truthful responses from agents (barring any behavioral biases or irrational behavior). In the

context of the market scoring rule, this means that players who are *myopic* and only look at the impact of their current report will report their true beliefs. Since agents cannot affect $s(\mathbf{q}_j)$ with their report, they seek to maximize $s(\mathbf{q}_{j+1})$, and if s is strictly proper, this involves a truthful report.

Definition 3.3. The **logarithmic scoring rule** is:

$$s(q, o_k) = b \ln(q_k).$$

In other words, your reward is some multiple of the natural log of the probability you assigned to the actual outcome.

Theorem 3.1. The logarithmic scoring rule is strictly proper.

Proof. Let your belief distribution be $(p_1, p_2, \ldots, p_k)$ over the outcome space. Your expected payoff from reporting $(q_1, q_2, \ldots, q_k)$ is $b \sum_{i=1}^{k} p_i \ln(q_i)$, where the q_i are subject to the restriction $\sum_{i=1}^{k} q_i = 1$. Dropping the constant, the Lagrangian is:

$$\mathcal{L} = \sum_{i=1}^{k} p_i \ln(q_i) + \lambda[1 - \sum_{i=1}^{k} q_i]$$

Taking the first-order conditions yields:

$$\frac{p_i}{q_i} - \lambda = 0 \quad \forall i$$

Thus, $q_i = \frac{p_i}{\lambda}$, and since $\sum_{i=1}^{k} q_i = 1$ and $\sum_{i=1}^{k} p_i = 1$, we know that $\lambda = 1$. This tells us that we maximize by reporting $q_i = p_i$, our true beliefs. This must be the maximum because there's only one critical point and the score is concave. □

In practice, a logarithmic market scoring rule is implemented using a **cost function based market maker**. The market maker offers one contract for each possible outcome, where the contract expires with value 1 if the outcome occurs and 0 if it does not occur. It tracks the net quantity $\mathbf{q} = (q_1, q_2, \ldots, q_n)$ of each contract that has been bought and uses a cost function $C(\mathbf{q})$ to determine the price at which contracts can currently be bought or sold. Specifically, to change the quantity from $\mathbf{q}_0$ to $\mathbf{q}_1$, a trader must pay $C(\mathbf{q}_1) - C(\mathbf{q}_0)$ (note that this amount could be negative, corresponding to the trader receiving a payment).

Definition 3.4. The LMSR market maker is a cost function based market maker with cost function

$$C(\mathbf{q}) = b \ln(\sum_{i=1}^{n} e^{\frac{q_i}{b}})$$

where b is a parameter chosen by the creator.

Definition 3.5. The **instantaneous price** for a cost function is

$$\pi_i(\mathbf{q}) = \lim_{h \to 0} \frac{C(\mathbf{q}_{-i}, q_i + h) - C(\mathbf{q}_{-i}, q_i)}{h} = \frac{\partial C}{\partial q_i}.$$

This is the price of buying an infinitesimally small amount of contract i. For the LMSR market maker,

$$\pi_i(\mathbf{q}) = \frac{e^{\frac{q_i}{b}}}{\sum_{i=1}^{n} e^{\frac{q_i}{b}}}.$$

The parameter b is important because it determines how sensitive prices are to changes in quantity. To see this intuitively, consider what happens to the instantaneous LMSR price if $b \to \infty$. Then all exponents will be ≈ 0, so the price will always be approximately $\frac{1}{n}$, reflecting a uniform distribution. On the other hand, if b is very small, then even a small increase in q_i will make $e^{\frac{q_i}{b}}$ very large compared to the other $e^{\frac{q_j}{b}}$, greatly increasing the instantaneous price.

This formulation has several nice properties:

- Trades can occur at any time. Traders do not need to submit orders and wait to find other traders willing to take the other side of the order. This can greatly increase liquidity.
- No arbitrage is possible by changing the price to a different value, then changing it back; your net payment is zero in this case.
- The sum of all instantaneous prices is always 1, so the prices can be interpreted as the market's current forecast of the probability of each outcome.
- Myopic risk-neutral traders have incentives to perform transactions that change the market forecast (instantaneous prices) to their true belief about probabilities. This trade is equivalent to reporting a true belief to the market scoring rule.

The equivalence of these two approaches might not be obvious at first glance. In the market scoring rule, it seems like agents are taking turns reporting beliefs, while with the market maker, it seems like they are buying and selling shares, making it more like a double auction. However, we can show that the following holds:

Theorem 3.2. Let $\vec{\pi}$ denote the instantaneous price *vector* for the LMSR market maker. If $\mathbf{q}$ is the net quantity of contracts sold by the LMSR market maker, then performing the transaction $\mathbf{h}$ with the market maker gives the same payoff as changing the belief report from $\vec{\pi}(\mathbf{q})$ to $\vec{\pi}(\mathbf{q} + \mathbf{h})$ using the logarithmic MSR.

Proof. With the market maker, buying $\mathbf{h}$ contracts when the market state is $\mathbf{q}$ results in a payoff of

$$h_k + C(\mathbf{q}) - C(\mathbf{q} + \mathbf{h})$$

when event k occurs. This can be written as:

$$h_k + b\left[\ln\left(\sum_{i=1}^{n} e^{\frac{q_i}{b}}\right) - \ln\left(\sum_{i=1}^{n} e^{\frac{q_i + h_i}{b}}\right)\right]$$

using the formula for C.

In the MSR, changing the belief report from $\vec{\pi}(\mathbf{q})$ to $\vec{\pi}(\mathbf{q}+\mathbf{h})$ when outcome k occurs results in a payoff of:

$$\begin{aligned} b\ln(\vec{\pi}(\mathbf{q}+\mathbf{h})_k) - b\ln(\vec{\pi}(\mathbf{q})_k) &= b\ln\left(\frac{e^{\frac{q_k+h_k}{b}}}{\sum_{i=1}^{n} e^{\frac{q_i+h_i}{b}}}\right) - b\ln\left(\frac{e^{\frac{q_k}{b}}}{\sum_{i=1}^{n} e^{\frac{q_i}{b}}}\right) \\ &= b\ln\left(\frac{e^{\frac{h_k}{b}}\sum_{i=1}^{n} e^{\frac{q_i}{b}}}{\sum_{i=1}^{n} e^{\frac{q_i+h_i}{b}}}\right) \\ &= h_k + b\left[\ln\left(\sum_{i=1}^{n} e^{\frac{q_i}{b}}\right) - \ln\left(\sum_{i=1}^{n} e^{\frac{q_i+h_i}{b}}\right)\right] \end{aligned}$$

using a few logarithm rules. Since the payoffs for any outcome from a transaction in one mechanism can be exactly replicated with a corresponding transaction in the other, the two are equivalent. □

4 Methodology

Our overall methodology consists of obtaining CDA data from actual markets run by the Good Judgement Project (GJP), then using a model to simulate the LMSR using belief reports from those same markets. As mentioned in the introduction, this approach allows for a comparison that comes much closer to isolating the effect of the mechanism than using market data from two different sources.

4.1 Dataset

Our data is obtained from the GJP's Year 3 prediction markets, where participants were required to report their true probability estimate for an event when submitting buy or sell orders. For example, in Figure 2, the second row represents a trader who wants to buy 20 contracts at a price of \$0.20, and who believes that there is a 35% chance of the relevant event occurring.

	timestamp	IFPID	User.ID	Op.Type	Order.ID	isBuy	isLong	Matching.Order.ID	Order.Price	Order.Qty	Trade.Price	Trade.Qty	Tru.Belief
0	7/31/2013 3:44	1128	3422	orderCreate	13.0	True	False	NaN	20.0	20.0	NaN	NaN	1.0
1	7/31/2013 5:07	1128	13012	orderCreate	64.0	True	True	NaN	20.0	20.0	NaN	NaN	35.0
2	7/31/2013 5:07	1128	13012	trade	64.0	True	True	13.0	20.0	0.0	20.0	20.0	NaN
3	7/31/2013 5:07	1128	3422	trade	13.0	True	False	64.0	20.0	0.0	20.0	20.0	NaN
4	7/31/2013 5:08	1128	13012	orderCreate	65.0	False	True	NaN	46.0	20.0	NaN	NaN	35.0

Figure 2: 5 sample trades from the GJP dataset

Each row of the data represents one of three operations: (1) an agent making a buy/sell order at a specified price and quantity; (2) an agent cancelling a previously created order; (3) two agents' orders being (fully or partially) matched, as described in Section 3.1.

In the data, there are a few instances of belief reports which are strongly inconsistent with their associated trades. For instance, there are orders to sell a contract at \$0.01 where the true belief of the probability of the event is reported as 99%. Orders like these are extremely unprofitable in expectation, so we suspect that they are due to user error or a misunderstanding of the question being asked. For the purposes of simulating the LMSR, trades which have a belief report more than 20 percentage points inconsistent with the order are ignored (i.e. buy orders where the belief is more than 20 p.p. less than the order price and sell orders where the belief is more than 20 p.p. greater than the order price). Rows that represent trades occurring are also ignored, because they correspond to the market operator matching bids and asks, rather than a new order and belief report being made.

We break down the questions asked by the GJP into three categories. There are (a) binary questions, (b) conditional binary questions, (c) categorical questions.

- Binary questions are questions with two answers, such as "Will Mahmoud Ahmadinejad resign or otherwise vacate the office of President of Iran before 1 April 2013?", with the choices Yes and No.

- Conditional binary questions are questions with two answers, but where several binary markets are open conditional on the outcome of another event. For example, there are two markets with the question "Will North Korea attempt launch of a multistage rocket between 7 January 2013 and 1 September 2013?", but one market is only valid in the case that the US announces additional sanctions against North Korea, while the other market is only valid in the case that this does not happen.
- Categorical questions have more than two choices. For example, "When will an Egyptian Referendum vote approve a new constitution?" with the choices (a) Between 1 Jul 2012 and 30 Sep 2012, (b) Between 1 Oct 2012 and 31 Dec 2012, (c) Between 1 Jan 2013 and 31 Mar 2013, (d) Event will not occur before 1 April 2013.

In performing our comparison, we focus on (a), as conditional binary and categorical questions may lead to statistical complications. We have 100 binary markets to work with.

4.2 Model

As mentioned in the previous section, the LMSR is really a belief report aggregation mechanism in that we expect agents to buy/sell some number of shares as a function of their true assessment of the probability of an event. In order to simulate the LMSR from the belief reports in our dataset, we have to choose a model for exactly how agents make this choice. In the most simple model, agents are risk-neutral and myopic, and so they perform the necessary operations to move the instantaneous price π to their true belief distribution. However, it seems a bit unlikely that real-world traders would be risk-neutral; most economic research suggests that people are risk-averse in almost all settings (in fact, some research shows that they are ridiculously risk-averse in small-stakes settings). Thus, we seek a model in which traders are risk-averse and at each trade, maximize their expected utility given their beliefs.

A model doing exactly this is proposed by Sethi and Vaughn [14]. Sethi and Vaughn consider an LMSR market maker with one security that expires with value \$1 if the relevant event occurs and \$0 otherwise (this is mathematically equivalent to the formulation in section 3). The market uses a cost function

$$C(q) = b \log(e^{\frac{q}{b}} + a)$$

where q is the total quantity of the contract that has been purchased, b is the liquidity parameter, and $a = \frac{1-p_0}{p_0}$ is determined by the market maker's prior p_0 such that the initial price $\pi(0) = C'(0) = p_0$.

The traders in this market are risk averse, with constant relative risk aversion (CRRA) utility

$$u(c) = \frac{c^{1-\rho}}{1-\rho}.$$

This is a standard utility function used in economics, where ρ measures how risk-averse traders are ($\rho \to 0$ represents risk-neutrality).

Each trader i has a portfolio $(y_{i,t}, z_{i,t})$ indicating the amount of cash and contracts, respectively, that they hold at time t. Agents are given an initial endowment of cash $y_{i,0}$ and have

$z_{i,0} = 0$ contracts initially. If trader i interacts with the automated market maker at time t, they choose a quantity $r_t \in \mathbb{R}$ of contracts to purchase, such that their expected utility given their current belief $p_{i,t}$ is maximized (this differs slightly from the original model, in which $p_{i,t}$ is assumed to be fixed over time for a given agent).

The exact way we run the model is as follows. We iterate over all trades in a given market and find the number of unique agents. For each agent, we initialize their portfolio with a certain amount of cash and 0 contracts. We then proceed through the belief reports in chronological order. For a given report, we identify the agent i who made the belief report and compute the number of shares they would purchase to maximize their expected utility. Given their portfolio $(y_{i,t-1}, z_{i,t-1})$, their current belief $p_{i,t}$, and the current market "state" (net contracts purchased) q_{t-1}, the agent maximizes

$$E[u(r_t)] = p_{i,t}u(z_{i,t-1}+y_{i,t-1}+r_t-(C(q_{t-1}+r_t)-C(q_{t-1})))+(1-p_{i,t})u(y_{i,t-1}-(C(q_{t-1}+r_t)-C(q_{t-1}))).$$

The first term represents the payoff if the event happens (and the value of the security is 1) and the second term represents the payoff if the event doesn't happen. After this optimal r_t is calculated, we update the current agent's portfolio and the market state:

$$\begin{aligned} y_{i,t} &= y_{i,t-1} - [C(q_{t-1} + r_t) - C(q_{t-1})] \\ z_{i,t} &= z_{i,t-1} + r_t \\ q_t &= q_{t-1} + r_t \end{aligned}$$

Note that we keep y and z the same for all other agents. The prediction generated at any point in time is simply

$$\pi(q_t) = C'(q_t) = \frac{e^{\frac{q_t}{b}}}{e^{\frac{q_t}{b}} + a}.$$

One thing to keep in mind is that the agent is subject to two constraints in choosing r_t:

- Agents can never have a short position that might require them to pay more cash than they have. Mathematically, $y_{i,t} + z_{i,t} \geq 0$.
- Agents are not allowed to spend more cash than they have in order to buy contracts. Mathematically, $y_{i,t} \geq 0$.

4.3 Implementation

We program this model in Python. For each real CDA market from the GJP, we simulate an LMSR market using the beliefs reported with a trade order as an agent's "true belief". Agents interact with the market maker in the same order in which they submitted trade orders in the CDA, but instead of submitting a trade order, they buy the amount of contracts that will maximize their expected utility based on their belief report. This is done by numerically computing the point where the derivative of expected utility is equal to zero (since CRRA utility is concave).

If an agent has belief report $p_{j,t}$, cash $y_{j,t-1}$, and owns $z_{j,t-1}$ contracts, and the total position

of the market is q_{t-1}, then the agent's expected utility from buying r_t contracts is:

$$p_{j,t}\frac{(z_{j,t-1}+y_{j,t-1}+r_t-C(q_{t-1}+r_t)+C(q_{t-1}))^{1-\rho}}{1-\rho}+(1-p_{j,t})\frac{(y_{j,t-1}-C(q_{t-1}+r_t)+C(q_{t-1}))^{1-\rho}}{1-\rho}$$

where the first term represents the utility from the event happening and the second term represents the utility from the event not occurring. Let

$$\begin{aligned}A(r_t) &= z_{j,t-1}+y_{j,t-1}+r_t-C(q_{t-1}+r_t)+C(q_{t-1})\\ B(r_t) &= y_{j,t-1}-C(q_{t-1}+r_t)+C(q_{t-1})\end{aligned}$$

Then the derivative of expected utility with respect to r_t is:

$$\begin{aligned}\frac{\partial E[u]}{\partial r_t} &= \frac{p_{j,t}}{1-\rho}(1-\rho)A^{-\rho}(1-C'(q_{t-1}+r_t))+\frac{1-p_{j,t}}{1-\rho}(1-\rho)B^{-\rho}(-C'(q_{t-1}+r_t))\\ &= p_{j,t}A^{-\rho}(1-\pi(q_{t-1}+r_t))+(p_{j,t}-1)B^{-\rho}\pi(q_{t-1}+r_t)\end{aligned}$$

where $\pi(q)=C'(q)$ represents the instantaneous price function. We can set this equal to zero numerically using Python's brentq solver.

To make sure that the value of r_t chosen is legal, we restrict the search range to $r_{min}<r<r_{max}$. r_{min} is the value for which $y_{i,t}+z_{i,t}=0$, representing how short (negative) your position can be before your losses could exhaust all of your cash. r_{max} is the value for which $y_{i,t}=0$, representing the number of contracts you can buy before using up all of your cash. Note that:

$$\begin{aligned}A(r_t) &= z_{j,t-1}+y_{j,t-1}+r_t-C(q_{t-1}+r_t)+C(q_{t-1})\\ &= y_{j,t}+z_{j,t}\\ B(r_t) &= y_{j,t-1}-C(q_{t-1}+r_t)+C(q_{t-1})\\ &= y_{j,t}\end{aligned}$$

Thus, r_{min} is a solution to $A(r)=0$ and r_{max} is a solution to $B(r)=0$.

One potential problem is that numerical root-finders require an input where a function is negative and another input where the function is positive (in order to guarantee a solution); it isn't certain whether the range (r_{min},r_{max}) satisfies this. However, we know that $\frac{\partial E[u]}{\partial r_t}>0$ as $r\to r_{min}$ because the first term goes to positive infinity (we are dividing by $A\to 0^+$, and $p(1-\pi)>0$) and the second term is finitely negative. We also know that $\frac{\partial E[u]}{\partial r_t}<0$ as $r\to r_{max}$ because the second term goes to negative infinity (we are dividing by $B\to 0^+$, and $p-1<0$) and the first term is finitely positive. This means that we can just use the solutions to $A(r)=0$ and $B(r)=0$ as our search range for the brentq solver, and that the maximum is guaranteed to be found.

4.4 Possible Criticism

One possible criticism of the model is that traders myopically maximize expected utility each time they trade; in reality, we might expect them to take their expectations about other traders'

behavior into account. For instance, a trader who wants to buy, but thinks recent events will cause a large amount of selling, might wait for the price to drop before buying. While there are theoretical results ([2], [3]) suggesting that truthful reporting is not always an equilibrium with the LMSR, empirical work by Jian and Sami [10] suggests that actual behavior might be more complex, especially in *unstructed* markets where traders are not forced to trade in a predetermined, commonly known order. In the absence of better understanding about equilibria in LMSR markets, myopic maximization seems like a reasonable baseline assumption.

Another criticism is that the beliefs traders reported in the CDA might not be the beliefs they would have had if the LMSR mechanism had been used. If beliefs are not only a function of outside information but are also influenced by the current market price and recent transactions, we might think that the different evolution of prices over time in the LMSR might cause beliefs to be slightly different. However, choosing a model for belief updating based on the price would necessarily entail making some assumptions (do traders take a linear combination of their prior and the price or use some other model?); without a broad consensus on how this occurs, it seems like preserving the belief reports from the CDA will still provide a useful baseline.

Also, one might note that we don't know the value of the liquidity parameter b or the coefficient of relative risk-aversion ρ, and our results might be influenced by our (somewhat arbitrary) choice of these parameters. To address these concerns, we compare performance for several different values of b and ρ.

5 Accuracy Comparison

To evaluate the accuracy of a prediction (i.e. the current market price), we mainly use **Brier score**.

Definition 5.1. If the predicted probability of an event happening is p and I_k is the indicator of whether the event actually occurred, then the Brier score is

$$(p - I_k)^2 + ((1 - p) - (1 - I_k))^2.$$

If the event occurs, this can be simplified to $2(1 - p)^2$, and if the event does not occur, this is equivalent to $2p^2$. The best score possible is 0, which is achieved if you assign probability 1 to the actual outcome (forecasting $p = 1$ when the event occurs or $p = 0$ when it does not) and the worst score possible is 2, which is achieved if you assign probability 0 to the actual outcome.

This is a well-known metric that has been used in previous evaluations of prediction market accuracy ([6], [7]) and is also a strictly proper scoring rule. To compute the Brier score associated with a given market, we take the price at the end of each day the market is open as a separate prediction. We then compute the Brier score of each end-of-day price and average these scores over the time the market was open (following [6]). Mathematically, let the score on day t for market i be $s_{i,t}$, and let market i be open for T_i days. We are computing:

$$\overline{s}_i = \frac{1}{T_i} \sum_{t=1}^{T_i} s_{i,t}.$$

While only using the final price may seem like a better indicator of a market's predictive accuracy, we argue that the day-average score is more useful for several reasons:

- We don't just care about the accuracy on the last day a market is open. If the market is horribly inaccurate for 99% of the time it is open and rapidly converges to the correct price 1 day before the event occurs, this is not as useful (though this is somewhat application-dependent) as a market which is slightly less accurate but is near the correct price almost the entire time it is open.
- Only looking at the price on the last day makes our results susceptible to noise. For instance, in an LMSR market, if the price of a very unlikely event (the Knicks winning the NBA championship) is close to zero but one irrational trader buys a huge number of shares at the last minute, the price may rise substantially, causing the market to look less accurate than it really was. Averaging over time doesn't completely ignore this noise, but smooths it out so that accuracy isn't overly penalized.

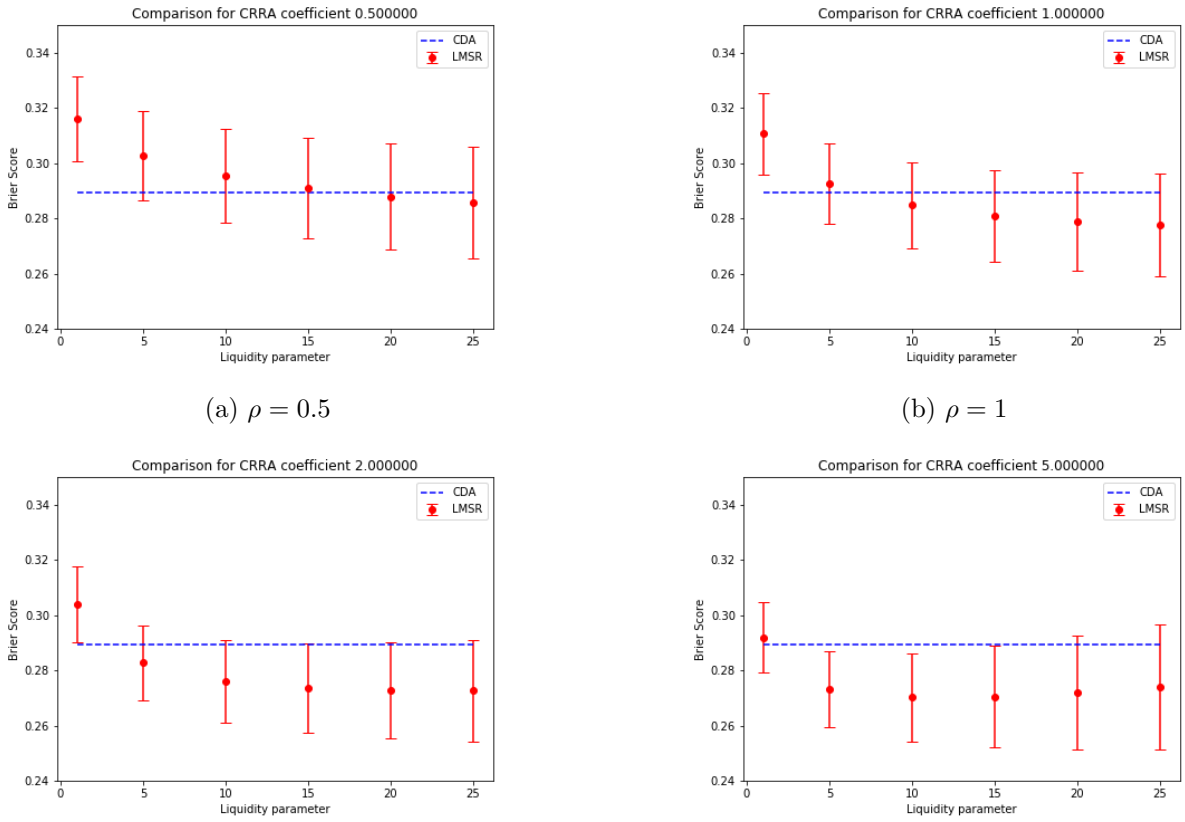
We also consider the percentage of markets in which one mechanism is more accurate than another. This can be computed by calculating the number of markets in which the day-average score of the CDA, $\overline{s}_{i,CDA}$, is lower than the day-average score of the LMSR, $\overline{s}_{i,LMSR}$.

5.1 Results: Uniform Cash

We compute the average Brier score for all binary CDA markets and their corresponding LMSR simulations, using parameter values $b \in \{1, 5, 10, 15, 20, 25\}$ and $\rho \in \{0.5, 1, 2, 5\}$. Note that the

CDA does not take in any parameters, so its predictions and Brier score remain the same for any parameter choice. We also set $y_{i,0} = 10$ for all agents, representing a hypothetical LMSR market where each agent is initially endowed with 10 units of cash (hence the title "uniform cash").

We indicate the CDA's average day-averaged Brier score across markets with a dotted line. For the LMSR, we obtain the day-average Brier score for each market and average over markets to get one observation for each parameter value. We plot this observation along with an error bar indicating two standard errors of the difference in Brier score (between the LMSR and CDA). If the average Brier score for the CDA is outside the error bars of the LMSR score, this (approximately) indicates that the difference in average scores is significant at the 5% level via a two-sided paired-t test. Exact values and significance levels for our t-tests can be found in the provided tables.



(a) $\rho = 0.5$

(b) $\rho = 1$

(c) $\rho = 2$

(d) $\rho = 5$

Figure 3: Comparison of LMSR and CDA Brier score (means with ± 2 standard errors of difference)

ρ \ b	1	5	10	15	20	25
0.5	3.38**	1.64	0.72	0.18	-0.15	-0.35
1.0	2.83**	0.43	-0.57	-0.98	-1.16	-1.23
2.0	2.06*	-0.94	-1.74	-1.89	-1.87	-1.78
5.0	0.41	−2.31*	−2.32*	−2.00*	-1.65	-1.32

Table 1: t-statistics when comparing LMSR and CDA performance for the uniform cash model; positive values mean the CDA was more accurate. * indicates significance at 5% level, ** indicates significance at 1% level.

On the whole, neither mechanism dominates the other. There are three combinations of (b, ρ) for which the CDA is more accurate at the 5% level, and there are three combinations for which the LMSR is more accurate at the 5% level. For the remaining 18 combinations, we fail to reject the null hypothesis that both mechanisms have equal average accuracy (across markets). In terms of simply comparing means, the LMSR has the lower mean Brier score in 16 parameter combinations, while the CDA has a lower score in the remaining 8. Higher risk aversion coefficients and higher liquidity parameters seem to be correlated with improved LMSR accuracy, though there appear to be diminishing returns to increasing b past 20 or so.

We also record the proportion of markets in which one mechanism outperforms the other for each parameter combination, and plot it in the heatmap below:

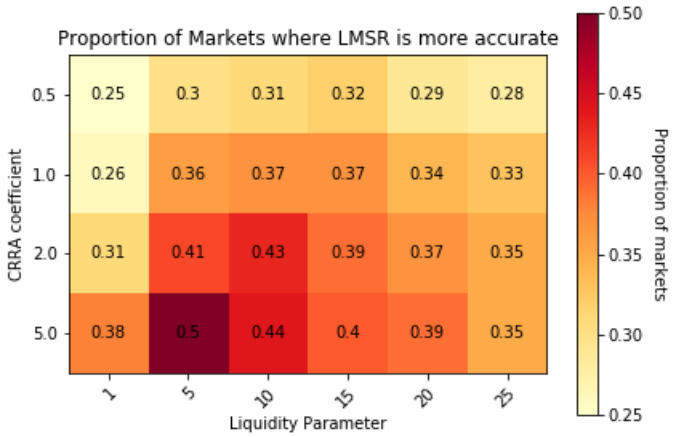


Figure 4: Plotting the proportion of markets in which the LMSR had a lower Brier score, for all parameter combinations.

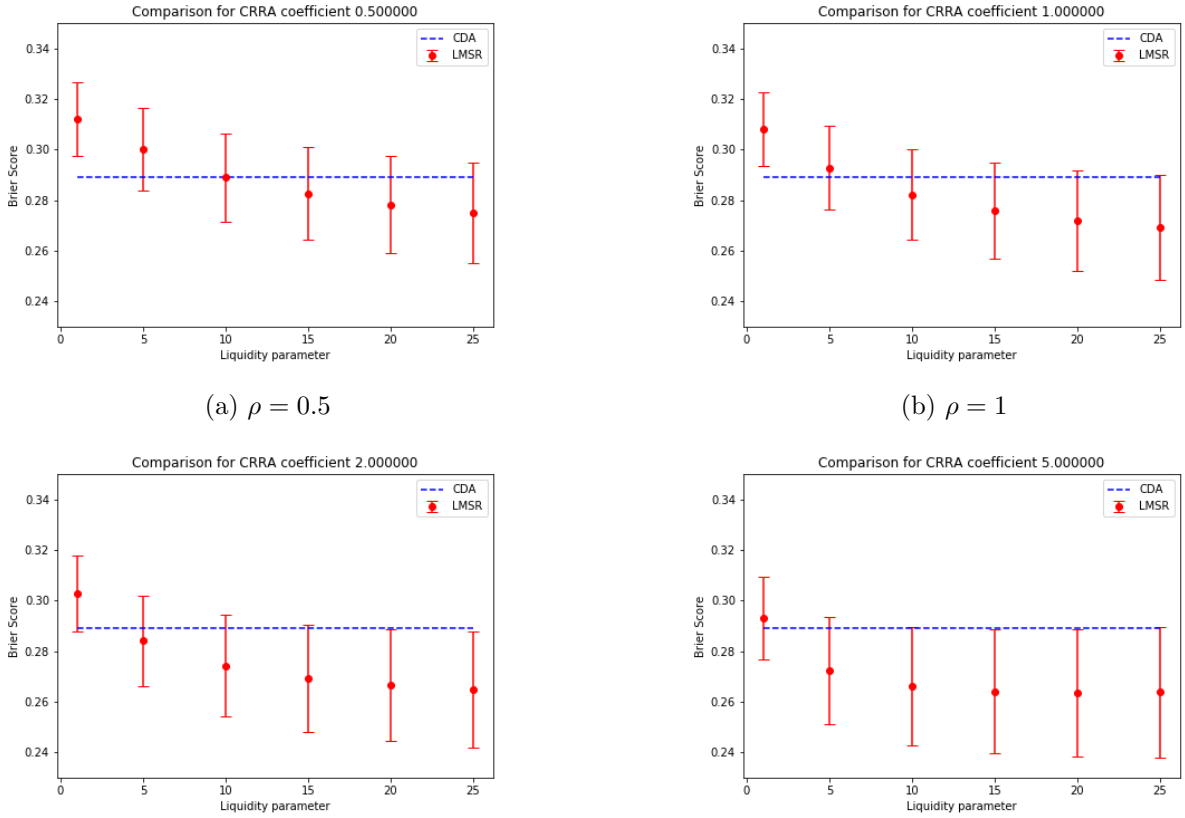
Interestingly, we see that the LMSR is outperformed by the CDA in a majority of markets for all parameter combinations except for $(b, \rho) = (5, 5)$, where they are tied. Reconciling this with the observation that the LMSR had a lower mean Brier score for many of these parameter combinations, it seems that the LMSR is sacrificing small losses in accuracy in a large number of markets for large accuracy gains in a small number of markets.

5.2 Results: Actual Cash

One problem with assuming that all agents start with the same amount of cash is that in the GJP's CDA markets, agents were given an initial endowment of cash and allowed to allocate it over *all* markets. In other words, an agent could choose to use 10% of their cash in one market, 30% in another, and 60% in a third market. This means that in any individual market, agents are willing to risk vastly different sums of money. Because the LMSR prediction (with risk-averse agents) depends on agents' budget constraints (the intuition is that with more money, you can afford to move the market price closer to your true belief), the "uniform cash" model might not be a totally accurate representation of what a hypothetical market with these traders would have looked like.

To address this issue, we estimate the budget each agent allocated to each market by computing how much money they would have needed to cover all of their submitted trades in that market. For a buy order, this represents how much money they would have needed to execute the buy order. For example, if an agent's only order in a market was to buy 50 shares at \$0.20, their budget is $50(0.2) = \$10$. For sell orders, we look at how much the agent would have lost if the event occurred. For example, if an agent made an order to short sell 50 shares at \$0.20, they would need $50(1 - 0.20) = \$40$ in case the event occurred. Finally, when an order is cancelled, we no longer count it against the agent's budget: if the agent submitted an order to buy \$10 of shares, then cancelled the order and submitted an order to buy \$20 of shares, their "necessary budget" would be \$20, not \$30.

After computing each trader's budget in each market, we run the simulation again, setting $y_{i,0}$ to our estimated values. Once again, exact values and significance levels for our t-tests can be found in the tables following the graphs.



(a) $\rho = 0.5$ (b) $\rho = 1$

(c) $\rho = 2$ (d) $\rho = 5$

Figure 5: Comparison of LMSR and CDA Brier score (means with ± 2 standard errors of difference)

ρ \ b	1	5	10	15	20	25
0.5	3.06**	1.29	-0.02	-0.7	-1.12	-1.4
1.0	2.52*	0.41	-0.76	-1.37	-1.7	-1.89
2.0	1.76	-0.56	-1.47	-1.84	−2.01*	−2.09*
5.0	0.48	-1.58	-1.95	−2.02*	−2.00*	-1.93

Table 2: t-statistics when comparing LMSR and CDA performance for the actual cash model; positive values mean the CDA was more accurate. * indicates significance at 5% level, ** indicates significance at 1% level.

The results are largely similar to the "uniform cash" case, but are slightly more favorable to the LMSR. The LMSR has a lower mean Brier score in 18 out of 24 parameter combinations, but it is only significantly better (at the 5% level) for 4 parameter combinations. This provides further evidence that on average, neither mechanism strictly dominates the other; this is true for a variety of risk-aversion coefficients and liquidity parameters, suggesting that market operators may not need to worry too much about agents' risk attitudes or precisely tuning b when creating a prediction market. As with the uniform cash case, having more risk-averse agents makes the LMSR more accurate, and having a higher liquidity parameter is also helpful, though there are

still diminishing returns.

We once again record the proportion of markets in which one mechanism outperforms the other for each parameter combination, and plot it in the heatmap below:

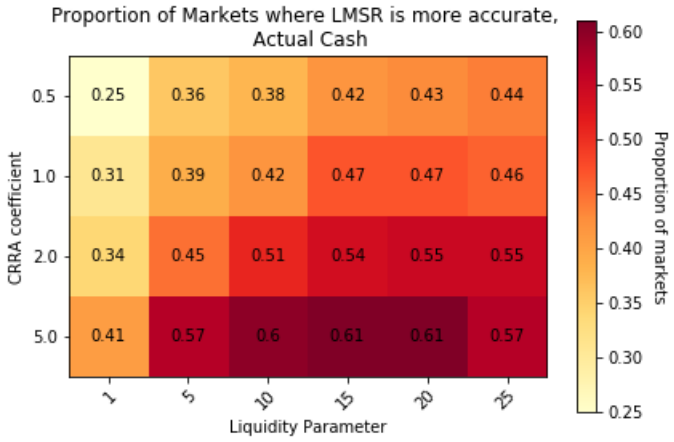


Figure 6: Plotting the proportion of markets in which the LMSR had a lower Brier score, for all parameter combinations.

Our results here are more consistent with the average Brier scores than in the uniform cash case. In most cases where the LMSR has a lower average Brier score, it is also more accurate in a majority of markets. However, there are still quite a few cases where the LMSR has a lower score in just 40-50% of markets, but still has a lower average Brier score. As in the uniform cash case, it seems that the LMSR is sacrificing small losses in accuracy in a large number of markets for large accuracy gains in a small number of markets.

5.3 Other Metrics

Even though the mechanisms seem to be similar in Brier score, we need to make sure that they are also comparable in other metrics. First, we make sure that both methods are well-calibrated. Calibration error measures how well predicted probabilities compare with actual probabilities (for example, that events predicted to occur 80% of the time actually occur around 80% of the time). To compute it, we divide up probabilities into several bins: $(0, 0.1), (0.1, 0.2), (0.2, 0.4)$, $(0.4, 0.6), (0.6, 0.8), (0.8, 1.0)$. The reason for splitting $(0, 0.2)$ into two bins is that many of our final prices are close to 0. Following the notation of Goel et al. [7], let $\tilde{p}_i$ be the midpoint probability of the bin in which market i falls. Then let $b_{\tilde{p}_i}$ be the fraction of markets in market i's bin for which the event actually occurred. If we have n markets, then our calibration error is

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\tilde{p}_i - b_{\tilde{p}_i})^2}.$$

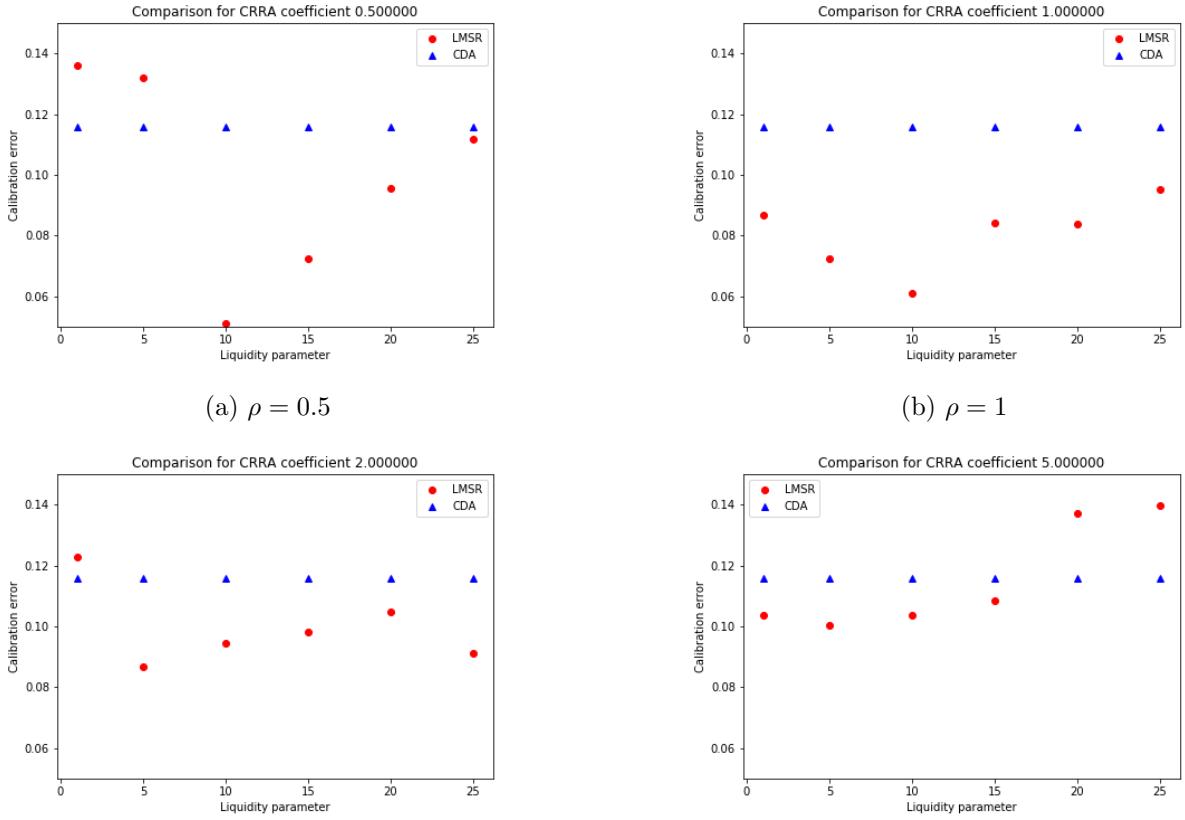
We also use discrimination, which penalizes uninformative predictions by measuring how much predicted probabilities vary. For instance, a method for predicting all MLB games that predicts a 50% win probability for each team in each game will necessarily have zero calibration error (all games go into the bin centered at 50%, and since each game results in one win and

one loss, the observed probability will also be 50%) but it will not be useful. Discrimination addresses this by measuring how much $b_{\tilde{p}_i}$ varies across bins. If b is the fraction of events that occur across all markets, then discrimination is

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(b_{\tilde{p}_i} - b)^2}.$$

Higher discrimination is generally better.

For each of our parameter combinations (b, ρ), we plot the calibration error and discrimination for the closing price in each market:



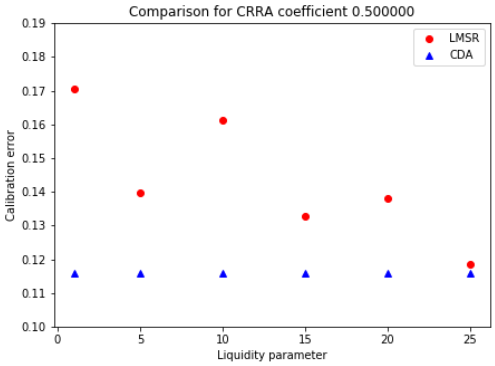
(a) $\rho = 0.5$

(b) $\rho = 1$

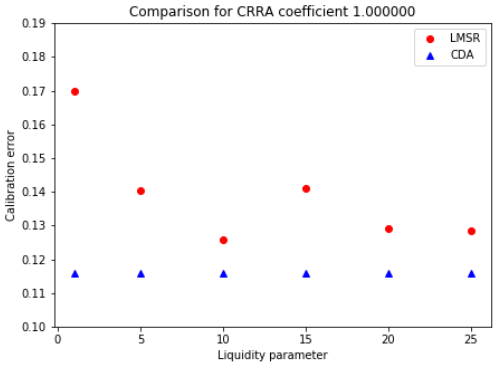
(c) $\rho = 2$

(d) $\rho = 5$

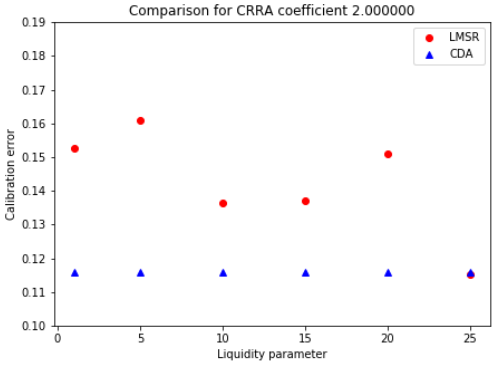
Figure 7: Comparison of LMSR and CDA calibration error, Uniform cash



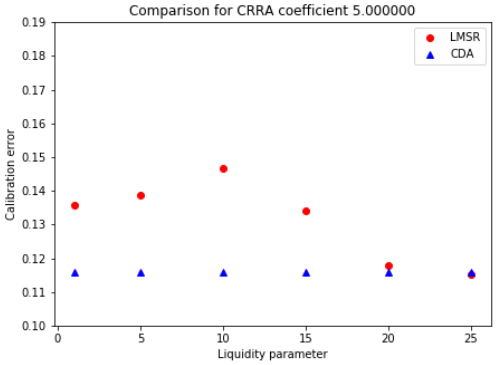
(a) $\rho = 0.5$



(b) $\rho = 1$

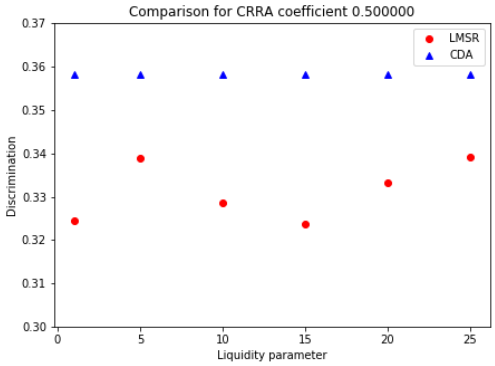


(c) $\rho = 2$

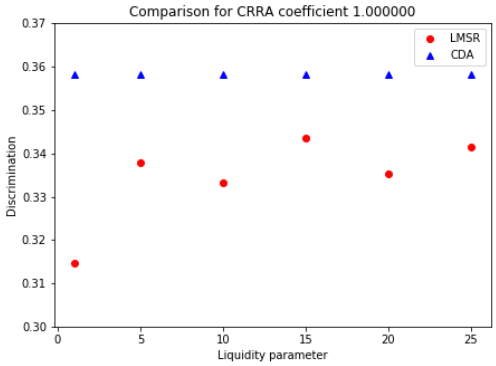


(d) $\rho = 5$

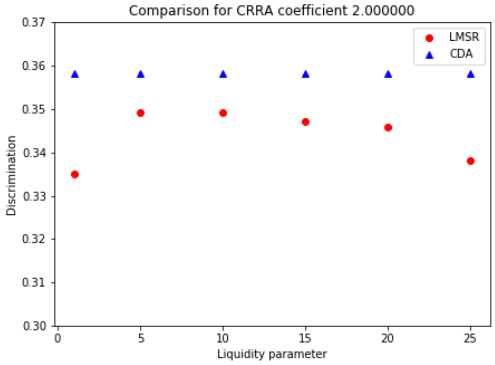
Figure 8: Comparison of LMSR and CDA calibration error, Actual cash



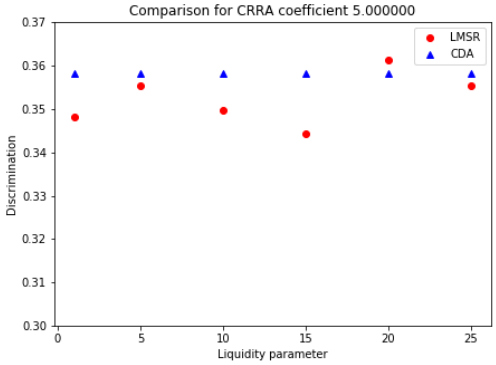
(a) $\rho = 0.5$



(b) $\rho = 1$

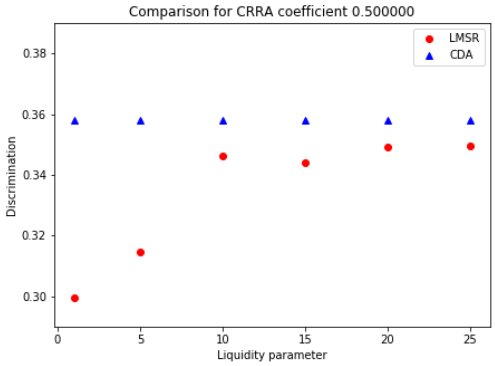


(c) $\rho = 2$

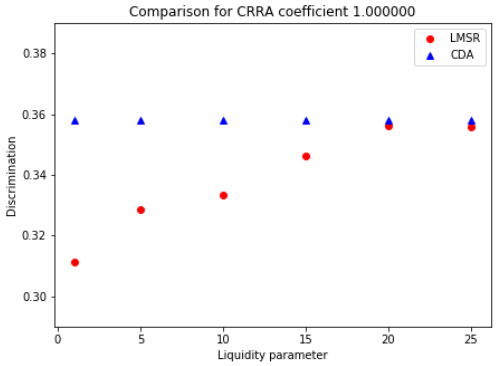


(d) $\rho = 5$

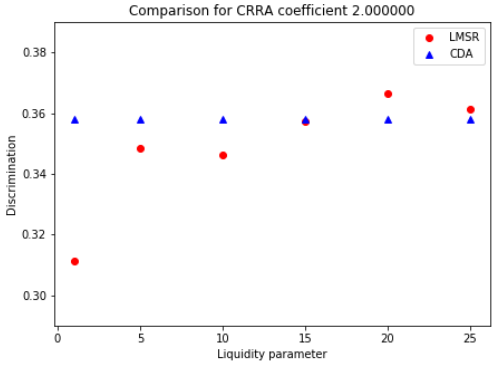
Figure 9: Comparison of LMSR and CDA discrimination, Uniform cash



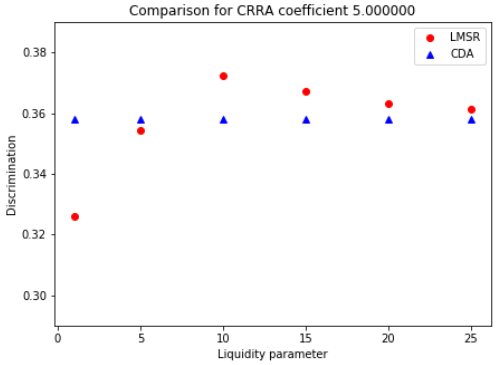
(a) $\rho = 0.5$



(b) $\rho = 1$



(c) $\rho = 2$



(d) $\rho = 5$

Figure 10: Comparison of LMSR and CDA discrimination, Actual cash

The CDA has higher calibration error for a majority (19 out of 24) of parameter values in the uniform cash simulation, but lower calibration error for a majority (22 out of 24) of parameter values in the actual cash simulation. For some combinations, the two mechanisms differ substantially (for instance, $(b, \rho) = (10, 0.5)$ in the uniform cash case); however, we don't have a notion of statistical significance for this metric, so it is difficult to conclude whether this is meaningful or not. This is especially true because our estimates can be a bit noisy due to small sample size. Going back to the example of $(b, \rho) = (10, 0.5)$, the 60% to 80% bin only has 1 market for the LMSR and 4 markets for the CDA. Our main takeaway is that there is no conclusive evidence suggesting one mechanism strictly dominates the other in terms of calibration.

In terms of discrimination, the CDA has higher discrimination for all but one parameter choice in the uniform cash simulation and all but six in the actual cash simulation. Despite this, there is never more than a 16.3% gap between the discrimination values of the two mechanisms. Again lacking a standard definition of statistical significance, we suspect that the CDA may be somewhat better with respect to discrimination, but can't guarantee that it offers a significant benefit.

5.4 Effect of Liquidity

It is important to distinguish from the two mechanisms having similar *average* accuracy across markets and the mechanisms having similar accuracy in any individual market. Taking one set of parameters ($\rho = 2, b = 10$), if we plot the difference in Brier score between mechanisms for each individual market, we obtain the following:

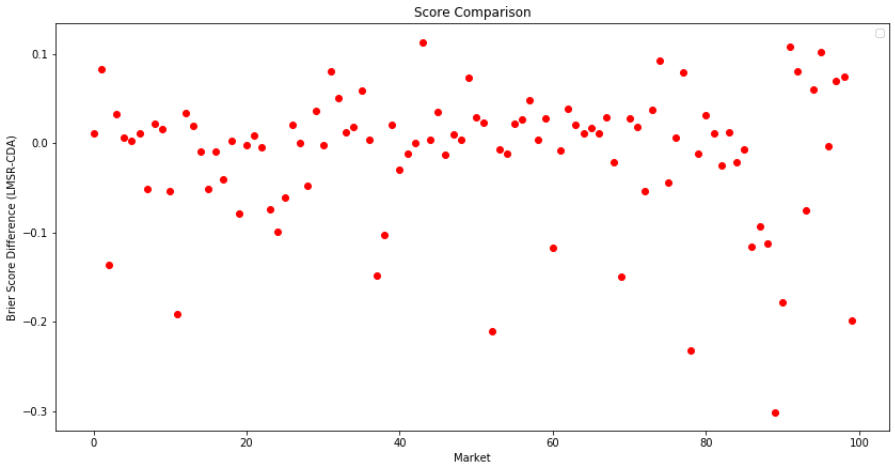


Figure 11: Plotting the difference in LMSR and CDA Brier score for each market

To get a sense of how large these differences are, note that the average CDA score across markets is 0.289. Even a score that is 0.05 lower represents a 17% improvement over the average. Thus, even though the markets generally have statistically similar accuracy on average, their performance can differ substantially for individual markets. One topic of interest might be exactly *when* one mechanism is more accurate than the other. This would allow market designers to choose which mechanism to use in a given situation and potentially increase overall accuracy.

The main variable that we suspect would be correlated with performance is liquidity. As we discussed in the background section, we know that CDA markets can fail to give meaningful predictions when liquidity is very low; this implies that the LMSR might be more accurate (have lower Brier scores) in less liquid markets. We compute the daily average bid-ask spread (the price of the lowest ask minus the price of the highest bid at the end of each day) of each market as a measure of liquidity, ignoring days where there are either no asks or no bids currently on the order book. We then regress the difference in LMSR and CDA Brier scores for each market on the spread of the market.

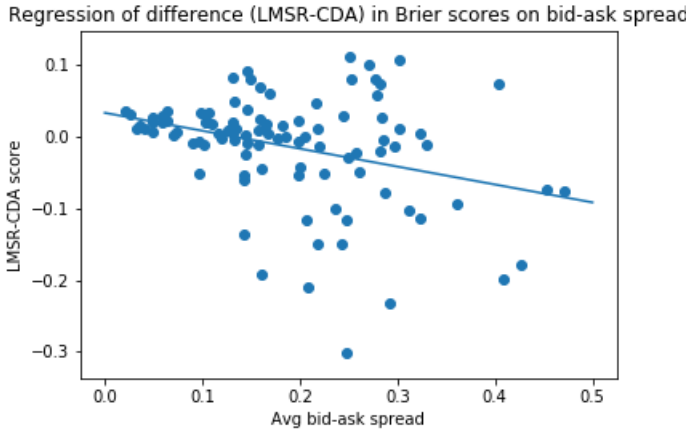


Figure 12: Regressing the difference in Brier score (LMSR-CDA) on bid-ask spread for $\rho = 2, b = 10$. This graph is fairly typical for all parameter values.

The coefficient on spread in this regression is significant at the 1% level for all possible parameter combinations, and the value of the coefficient ranges from -0.22 to -0.36. One sample regression is plotted above for visualization. The significantly negative slope indicates that the LMSR has a lower score (and is more accurate) in markets with a higher spread (less liquidity). We suspect that by using the LMSR in markets that are less liquid and the CDA in markets that are more liquid, we will obtain significant gains in accuracy.

6 Aggregating the Algorithms

Another way to utilize the relationship between liquidity and accuracy is to consider what will happen if we run *both* mechanisms simultaneously and combine their predictions in a way that is dependent on liquidity. This would give us a "boosted" or "hybrid" algorithm that may give significantly better predictions than either individual mechanism on average.

6.1 Boosted Algorithm

Our proposed method is to take a linear combination of the two mechanisms' predictions, where the coefficient of the linear combination depends on liquidity. Let $p_{LMSR,i,t}$ and $p_{CDA,i,t}$ represent the prediction made by each mechanism in market i on day t. Furthermore, let d_i represent the daily average bid-ask spread in market i. To generate the boosted prediction in market i on day t, we take a linear combination

$$p_{i,t} = \alpha(d_i) \cdot p_{LMSR,i,t} + (1 - \alpha(d_i)) \cdot p_{CDA,i,t}$$

for the final prediction on each day, where $\alpha(d) = \Phi(b_0 + b_1 \cdot d)$.

The reason for the probit function is that it forces the coefficient to be between 0 and 1; this prevents our aggregated prediction $p_{i,t}$ from being outside the range defined by $[\min(p_{LMSR,i,t}, p_{CDA,i,t}), \max(p_{LMSR,i,t}, p_{CDA,i,t})]$. While this has the slight downside of preventing extremization, it gives our coefficient an intuitive interpretation: a coefficient closer to 1 implies that we should trust the LMSR prediction more, while a coefficient close to 0 implies that we should trust the CDA prediction. It could also be viewed as an estimate of the probability that the LMSR prediction will be more accurate than the CDA prediction, given the liquidity of the market.

The parameters of the model are (b_0, b_1), which capture the effect of the spread on α. To estimate these values, we create a loss function and minimize it with respect to our parameters. Our choice of loss function is the total day-average Brier score across all markets for a given choice of (b_0, b_1); we can write this as:

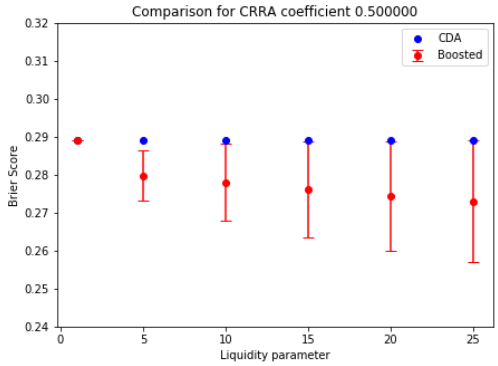
$$\begin{aligned}
S(b_0, b_1) &= \sum_{i=1}^{N} \frac{1}{T_i} \sum_{t=1}^{T_i} s_{i,t} \\
&= \sum_{i=1}^{N} \frac{1}{T_i} \sum_{t=1}^{T_i} 2(p_{i,t} - I_k)^2 \\
&= \sum_{i=1}^{N} \frac{1}{T_i} \sum_{t=1}^{T_i} 2(\alpha(d_i) \cdot p_{LMSR,i,t} + (1 - \alpha(d_i)) \cdot p_{CDA,i,t} - I_k)^2 \\
&= \sum_{i=1}^{N} \frac{1}{T_i} \sum_{t=1}^{T_i} 2(\Phi(b_0 + b_1 \cdot d_i) \cdot p_{LMSR,i,t} + (1 - \Phi(b_0 + b_1 \cdot d_i)) \cdot p_{CDA,i,t} - I_k)^2
\end{aligned}$$

Since all other quantites are known, we can numerically compute the value of (b_0, b_1) that will minimize the loss.

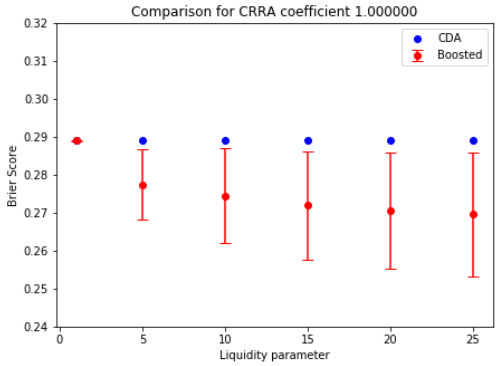
6.2 Results

To test whether this approach is promising, we divide up our 100 markets into test and training sets and perform cross-validation for each (b, ρ) combination. We perform ten rounds of ten-fold cross validation. In each round, we randomly divide the markets into ten groups of 10; we then iterate through the groups, using one group as test data and the other nine groups as training data that we compute our optimal (b_0, b_1) from. Using this (b_0, b_1), we compute the boosted predictions in both the test and training markets. In each of the ten rounds, we choose a different random partition of the 100 markets so that our results are robust to any correlations that might be present in the ordering of the markets.

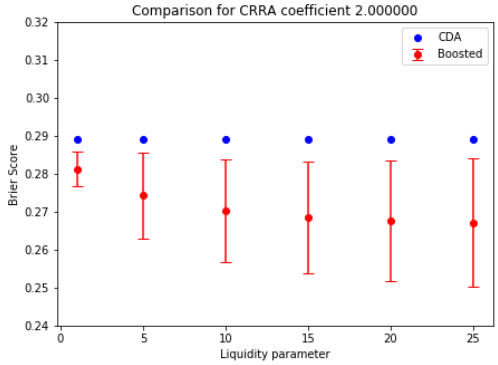
First, we display the results from our training data. In each round of cross-validation, a given market is used nine times as training data, so we have 90 boosted training predictions for each market across the ten rounds. We average these predictions to get one training prediction for each of our 100 markets. As in the previous section, we plot the average (across markets) Brier score from the boosted algorithm, and test whether it is significantly better than the CDA or the LMSR through a two-sided paired-t test. For this set of simulations, the LMSR predictions that we use are from the uniform cash model.



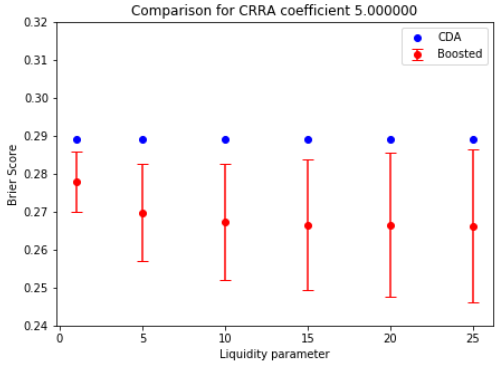
(a) $\rho = 0.5$



(b) $\rho = 1$

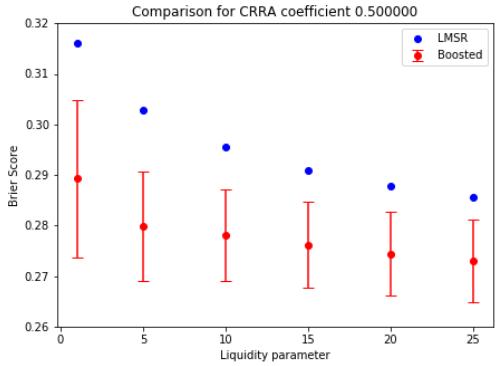


(c) $\rho = 2$

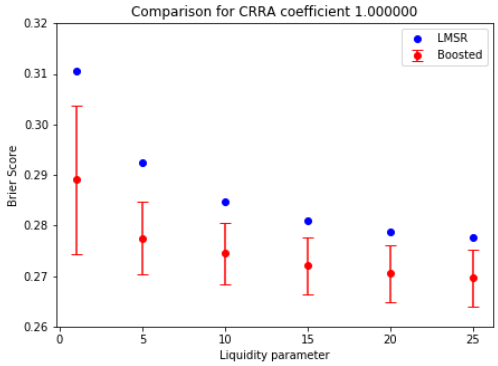


(d) $\rho = 5$

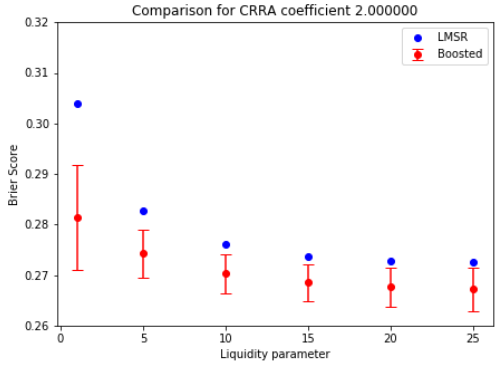
Figure 13: Comparison of CDA and boosted Brier score on training data (means with ± 2 standard errors of difference)



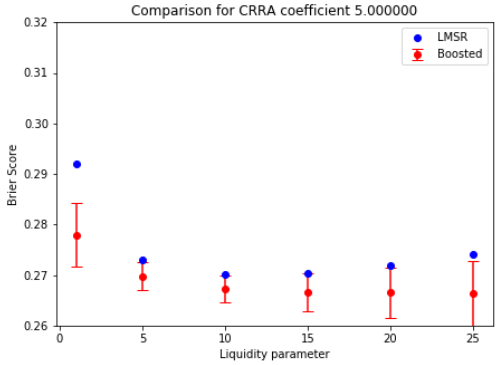
(a) $\rho = 0.5$



(b) $\rho = 1$



(c) $\rho = 2$



(d) $\rho = 5$

Figure 14: Comparison of LMSR and boosted Brier score on training data (means with ± 2 standard errors of difference)

ρ \ b	1	5	10	15	20	25
0.5	-1.375	2.83^{**}	2.153^{*}	2.038^{*}	2.00^{*}	1.981^{*}
1.0	3.881^{**}	2.485^{*}	2.318^{*}	2.368^{*}	2.381^{*}	2.363^{*}
2.0	3.495^{**}	2.591^{*}	2.762^{**}	2.737^{**}	2.657^{**}	2.562^{*}
5.0	2.813^{**}	2.991^{**}	2.814^{**}	2.571^{*}	2.353^{*}	2.225^{*}

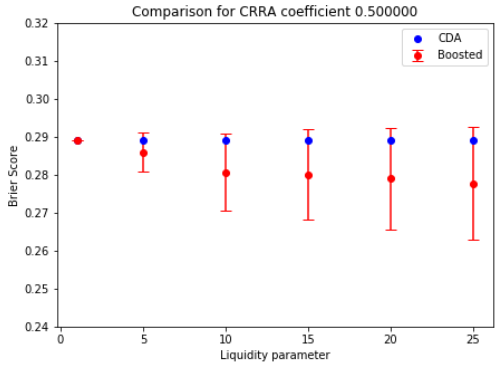
Table 3: t-statistics when comparing CDA and boosted algorithm performance on training data; positive values mean the boosted algorithm was more accurate. * indicates significance at 5% level, ** indicates significance at 1% level.

ρ \ b	1	5	10	15	20	25
0.5	3.384**	4.116**	3.8**	3.397**	3.168**	3.027**
1.0	2.878**	4.073**	3.351**	3.079**	2.881**	2.79**
2.0	4.247**	3.476**	2.925**	2.75**	2.666**	2.42*
5.0	4.267**	2.426*	2.077*	2.031*	2.124*	2.348*

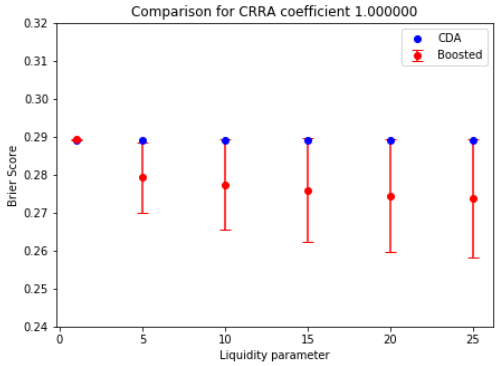
Table 4: t-statistics when comparing LMSR and boosted algorithm performance on training data; positive values mean the boosted algorithm was more accurate. * indicates significance at 5% level, ** indicates significance at 1% level.

Our results indicate that we have improved accuracy significantly in almost all cases. The hybrid algorithm is better than the CDA at the 5% level for 23 out of 24 parameter combinations, and better than the LMSR at the 5% level for all 24 parameter combinations (while also being better at the 1% level in 18 combinations). This validates our hypothesis that liquidity is a major factor in determining which method is more accurate. Had we chosen a linear combination based on some uncorrelated factor, we would not have expected significant improvement.

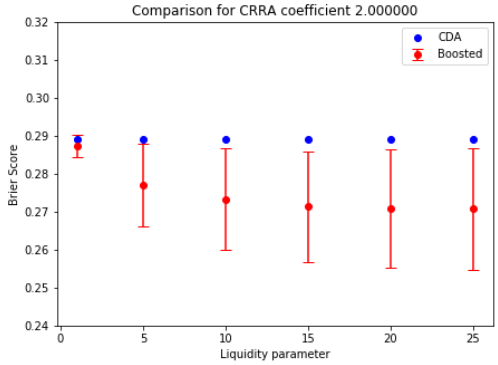
Of course, the ultimate test in comparing prediction accuracy is on actual test data. In each round of cross-validation, a given market is used once as test data, so we have 10 boosted test predictions for each market in total. We average these predictions for each of our 100 markets, and compare the average Brier scores of these test predictions against the Brier scores from using the CDA or the LMSR by itself. We display our results from our test data below. In our tables, we also indicate significance at the 10% level, as this would be equivalent to significance at the 5% level in a one-sided test.



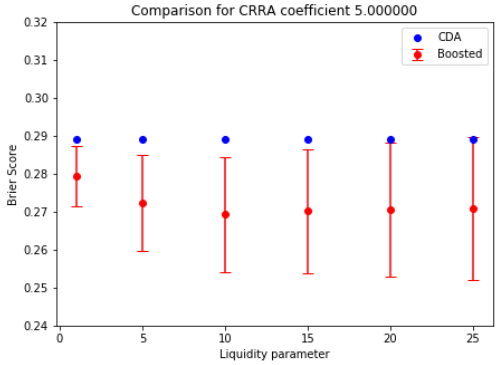
(a) $\rho = 0.5$



(b) $\rho = 1$

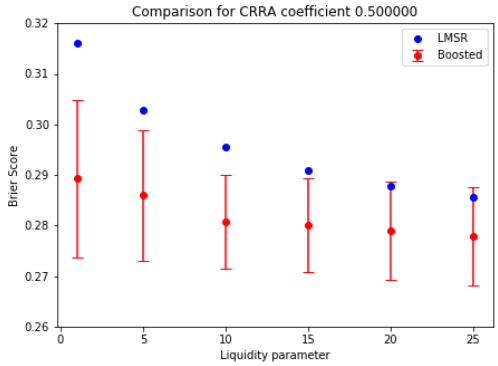


(c) $\rho = 2$

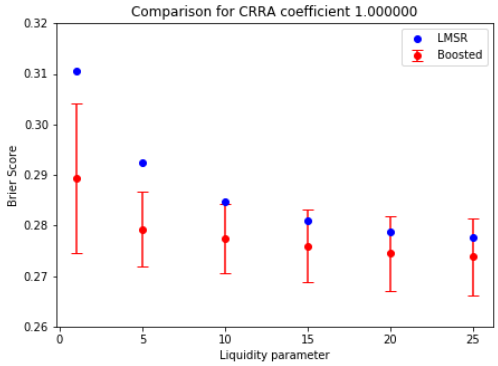


(d) $\rho = 5$

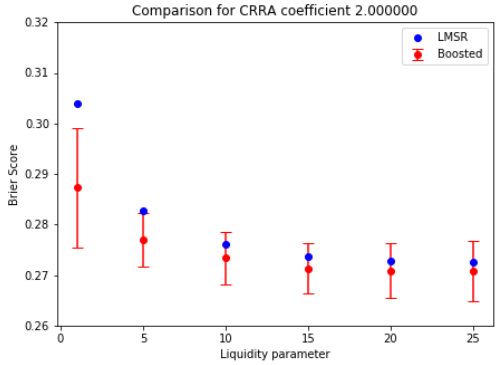
Figure 15: Test data comparison of CDA and boosted Brier score (means with ± 2 standard errors of difference)



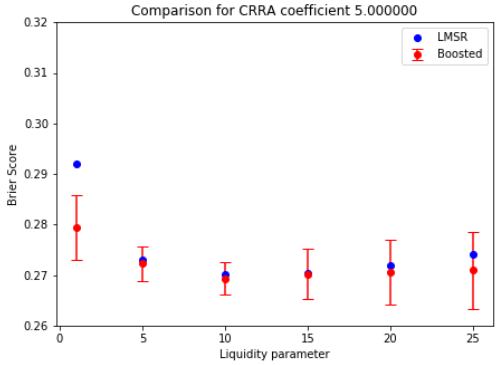
(a) $\rho = 0.5$



(b) $\rho = 1$



(c) $\rho = 2$



(d) $\rho = 5$

Figure 16: Test data comparison of LMSR and boosted Brier score (means with ±2 standard errors of difference)

ρ \ b	1	5	10	15	20	25
0.5	-1.272	1.248	1.662^{+}	1.499	1.498	1.519
1.0	-0.426	2.114^{*}	1.955^{+}	1.895^{+}	1.957^{+}	1.933^{+}
2.0	1.333	2.192^{*}	2.341^{*}	2.404^{*}	2.318^{*}	2.253^{*}
5.0	2.462^{*}	2.635^{**}	2.567^{*}	2.277^{*}	2.057^{*}	1.907^{+}

Table 5: t-statistics when comparing CDA and boosted algorithm performance; positive values mean the boosted algorithm was more accurate. $^{+}$ indicates significance at 10% level, * indicates significance at 5% level, ** indicates significance at 1% level (for a two-sided test).

ρ \ b	1	5	10	15	20	25
0.5	3.384^{**}	2.543^{*}	3.111^{**}	2.282^{*}	1.768^{+}	1.587
1.0	2.83^{**}	3.511^{**}	2.112^{*}	1.365	1.154	0.987
2.0	2.748^{**}	2.099^{*}	1.025	0.935	0.687	0.595
5.0	3.841^{**}	0.505	0.519	0.094	0.39	0.791

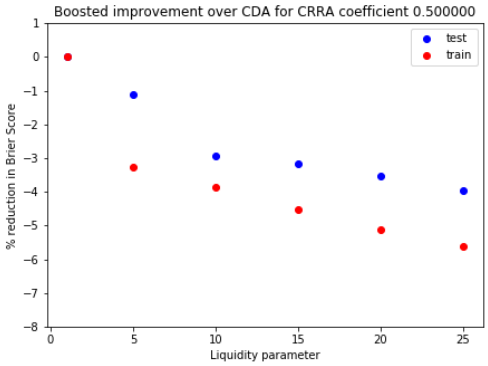
Table 6: t-statistics when comparing LMSR and boosted algorithm performance; positive values mean the boosted algorithm was more accurate. $^{+}$ indicates significance at 10% level, * indicates significance at 5% level, ** indicates significance at 1% level (for a two-sided test).

The boosted algorithm is more accurate than the CDA at the 5% level for 11 parameter combinations, and is more accurate at the 10% level (significant in a one-sided test) for six additional combinations. It is also more accurate than the LMSR at the 5% level for 10 parameter combinations, and is more accurate at the 10% level for one additional combination. Overall, in 23 out of 24 parameter combinations, the hybrid algorithm outperforms at least one mechanism at the 10% level (this holds for 18 out of 24 using the stricter 5% standard).

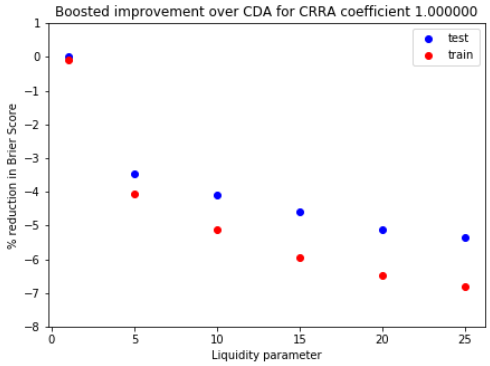
In addition to generally performing significantly better than the worse of the two mechanisms, the boosted algorithm performs at least as well as the better of the two mechanisms. The hybrid algorithm is only less accurate than either mechanism for two parameter combinations (it is less accurate than the CDA for $(\rho = 0.5, b = 1)$ and $(\rho = 1, b = 1)$), and the difference is insignificant in both cases. Furthermore, we observe that the mean Brier score is essentially the same as the CDA in these cases.

Something slightly disappointing is that while the hybrid algorithm outperformed both mechanisms simultaneously for 22 out of 24 parameter combinations in the test data, it was only significantly better than both mechanisms at the 5% level for 3 parameter combinations (as compared to 23 out of 24 parameter combinations in the training data). While this might seem like a sign of overfitting, we note that the t-statistics were often quite close to the 5% cutoff (especially for the CDA) in the training data; thus, even a small drop-off in performance would lead to statistical insignificance.

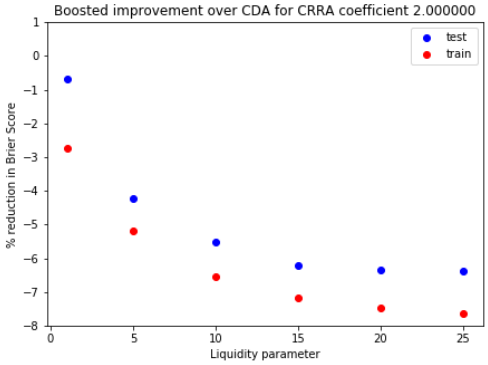
To provide further evidence for our argument that we are not overfitting, we plot the percent reduction in Brier score on training and test data for all parameter values:



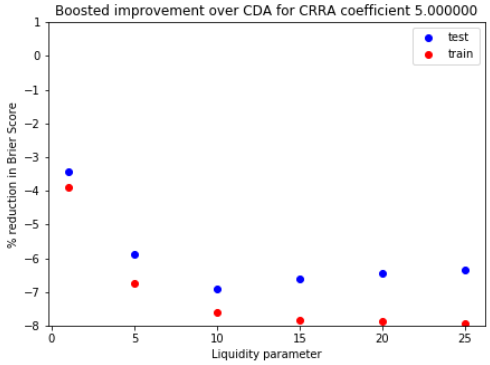
(a) $\rho = 0.5$



(b) $\rho = 1$

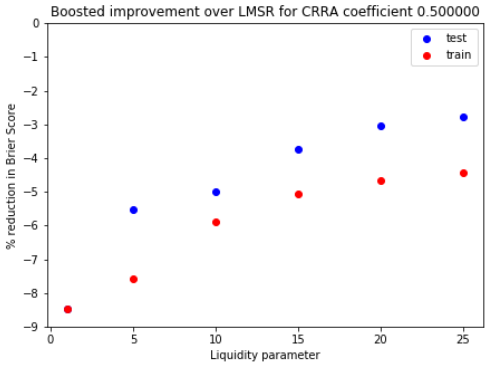


(c) $\rho = 2$

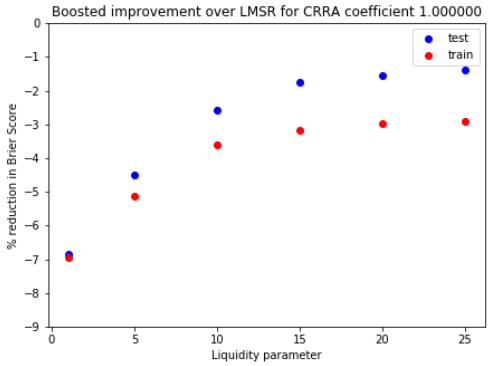


(d) $\rho = 5$

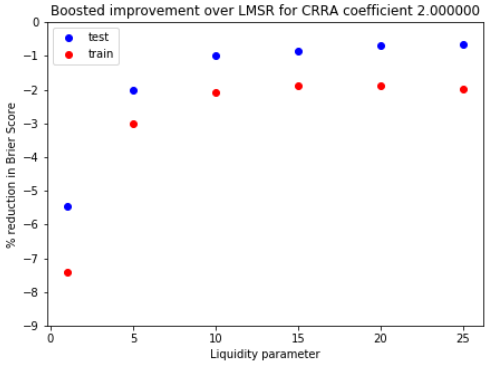
Figure 17: Comparing % decrease in Brier score from boosted algorithm vs. CDA for test and training data.



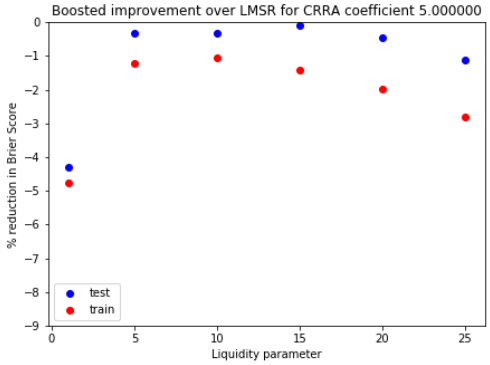
(a) $\rho = 0.5$



(b) $\rho = 1$



(c) $\rho = 2$



(d) $\rho = 5$

Figure 18: Comparing % decrease in Brier score from boosted algorithm vs. LMSR for test and training data.

From the graphs, we see that while the algorithm reduces Brier score more in training data, the difference in performance between training and test data is usually just 1-2 percentage points. On average, across all parameter values, the boosted algorithm outperforms the CDA by 4.26% on test data and 5.39% on training data. It outperforms the LMSR by 2.69% on test data and 3.84% on training data.

We also think it is helpful to consider whether the reduction in Brier score is *practically* significant, even though it is not always statistically significant. If a proposed algorithm achieves an average Brier score improvement of 0.1% that is highly significant, it might have less potential than one that improves Brier score by 5% but is only borderline significant due to a relatively small sample size. The following table shows the percentage reduction in Brier score achieved by the hybrid mechanism over the CDA and LMSR in test data for each parameter choice:

ρ \ b	1	5	10	15	20	25
0.5	$(0, -8.47)$	$(-1.13, -5.53)$	$(-2.94, -4.99)$	$(-3.17, -3.73)$	$(-3.53, -3.04)$	$(-3.96, -2.76)$
1.0	$(0.01, -6.86)$	$(-3.45, -4.51)$	$(-4.10, -2.58)$	$(-4.59, -1.76)$	$(-5.12, -1.56)$	$(-5.34, -1.38)$
2.0	$(-0.70, -5.45)$	$(-4.23, -2.02)$	$(-5.50, -0.98)$	$(-6.20, -0.86)$	$(-6.35, -0.70)$	$(-6.39, -0.66)$
5.0	$(-3.44, -4.31)$	$(-5.88, -0.32)$	$(-6.89, -0.31)$	$(-6.60, -0.09)$	$(-6.45, -0.47)$	$(-6.33, -1.12)$

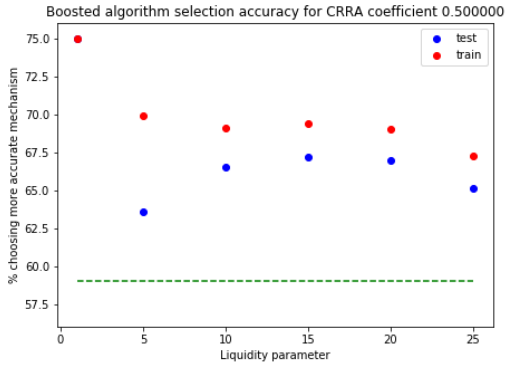
Table 7: Ordered pairs showing (% change in Brier score vs. CDA, % change in Brier score vs. LMSR).

Overall, the results are quite parameter-dependent. While there are some parameter combinations for which the boosted algorithm simultaneously outperforms both mechanisms by 3% or more, there are also some parameter combinations for which the boosted predictions outperform one mechanism by around 5% but only beat the other by 1% or less. It seems that the boosted algorithm has potential for meaningful simultaneous improvements over both mechanisms, but this potential is strongly dependent on the relative performance of the CDA and LMSR. Referring back to Table 1, for parameter values where the CDA does significantly better than the LMSR, the boosted algorithm struggles to substantially improve on the CDA, and for parameter values where the LMSR does significantly (or almost significantly) better than the CDA, the boosted algorithm struggles to substantially improve on the LMSR.

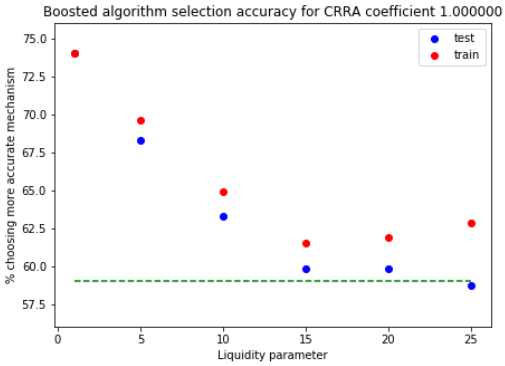
Despite this, it seems that given markets run using the CDA and the LMSR, our hybrid approach is a good way to virtually guarantee (without any user input!) that we will get predictions as good as the better of the two mechanisms on average. This is important, because a priori it would be hard to identify which mechanism would be better for a given application (recall that we are using test data, so we don't know this). Further improvements beyond the better mechanism are generally achieved, but are not always statistically or practically significant.

6.3 Results: Identification of More Accurate Mechanism

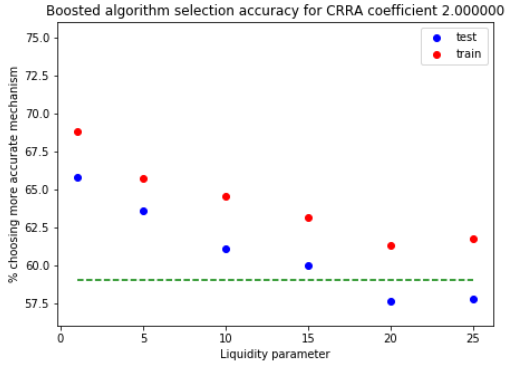
We also check whether the boosted algorithm is generally identifying the more accurate mechanism and placing more weight on it. The exact criterion we use is that when a coefficient α_i is generated for a market, we check if $\alpha_i > 0.5$ (more weight on the LMSR) when the LMSR is more accurate and if $\alpha_i < 0.5$ (more weight on the CDA) when the CDA is more accurate. We then compute the proportion of test data markets for which α_i satisfies this criterion. Because we only have 100 unique markets, and are using each one as test data 10 times, we test if we are doing better than random by converting our accuracy to a rate per 100 markets. We then compare it to a binomial distribution with $n = 100, p = 0.5$. In our plots, we draw a line at 59 as an indicator of the cutoff for significance at the 5% level for a one-sided test.



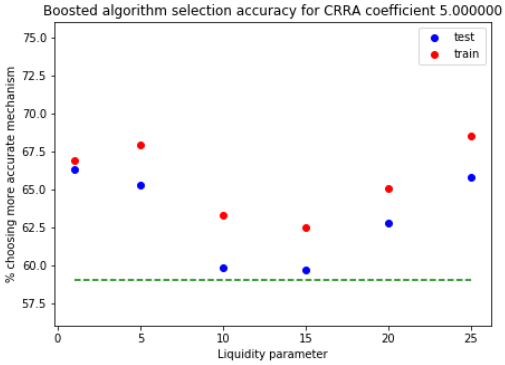
(a) $\rho = 0.5$



(b) $\rho = 1$



(c) $\rho = 2$



(d) $\rho = 5$

Figure 19: Rate at which boosted algorithm selects more accurate mechanism in test and training data.

These results are more favorable in terms of significance; the boosted algorithm is significantly better than random (at the 5% level) at selecting the more accurate mechanism in 21 out of 24 parameter combinations. Once again, while we generally see declines in performance from training to test data, it is usually on the order of a few percentage points or less, providing more evidence that we are not overfitting.

7 Discussion

One conclusion we draw from our results in Section 5 is that the LMSR (even our simplified model of it) has the potential to be just as accurate as the CDA on average. The result that the average Brier scores are not significantly different from each other with the vast majority of parameter combinations suggests that in most settings, we might be relatively indifferent between the two mechanisms. We think this result is especially robust because there is no bias introduced by different trader populations or different market features (for example, how long the market is open or how many traders there are). Furthermore, our results are derived from markets run on a realistic question in a non-laboratory setting and are not affected by extremely low liquidity or combinatorial outcome spaces.

This result contrasts somewhat with Ledyard et al. and Healy et al., which found that the LMSR outperformed the CDA in accuracy in "complex" settings. However, it is more consistent with the observations of Dana et al., as we have also found that a belief aggregation mechanism (the LMSR) can perform just as well as CDA market prices in terms of Brier score.

The similarity in accuracy seems to be driven by two offsetting effects. While the CDA is often more accurate than the LMSR in a slight majority of markets, the LMSR is able to overcome this disadvantage by having large accuracy gains in those markets where it is more accurate. One reason why this might occur is that risk-aversion and budget constraints play a large role in the LMSR; it could be the case that LMSR prices are not as extreme as CDA prices in markets where the outcome is very obvious from the start. This would result in small penalties in a large number of markets, but it could benefit the LMSR greatly in a market where the outcome is less certain.

With regards to parameter choices, we find that the LMSR generally improves relative to the CDA as agents become more risk-averse. This suggests that market creators may wish to operate LMSR markets if they know their agents are especially risk-averse and CDA markets if they suspect their agents are relatively risk-neutral. We also find that increasing the liquidity parameter makes the LMSR more accurate up to a certain point; the best values were $b = 10, 15$ for the "uniform cash" model and $b = 15, 20$ for the "actual cash" model. Though the literature contains relatively little concrete knowledge about how to set an optimal b (in fact, it is referred to as "more art than science" [4]), one takeaway is that there does seem to be a sweet spot for b that maximizes accuracy.

An important caveat is that our comparison uses a simulated LMSR market; it's entirely possible that a real LMSR market (or a more sophisticated model) could have even better accuracy. Thus, we might consider our results a demonstration of the potential accuracy the LMSR mechanism could achieve. One example of how a real market could differ is that in a true LMSR market, agents would see the current price and could update their beliefs accordingly, diluting the impact that outliers have on our current results. Another interpretation of our results is that a mechanism that *simulates* the LMSR given belief reports from risk-averse agents can aggregate information just as well as the CDA. This could be useful on its own as another belief aggregation algorithm, similar to the algorithm considered by Dana et al.

From Section 6, we conclude that a relatively simple hybrid mechanism can virtually guarantee performance on par with the better of the CDA and LMSR on test data, where we don't know which mechanism will be more accurate. It also has the potential to meaningfully outperform both mechanisms, even on test data. This is an encouraging result, as it indicates that we can leverage the strengths of both mechanisms to build a hybrid mechanism that is better on average without overfitting. Our conclusion is similar to that of Dana et al., who find that the simple average of "Prices" (CDA prices) and "Beliefs" (their algorithm applied to belief reports) is significantly more accurate than Prices alone. Going forward, prediction market operators may wish to simultaneously operate LMSR and CDA markets on events (alternatively, they could operate only the CDA but request belief reports). The fact that this improvement was largely preserved in cross-validation indicates that they could estimate values of (b_0, b_1) on some training markets, then apply the algorithm to other markets, and still achieve accuracy gains.

We now take some time (and space) to address possible concerns and criticisms:

1. The model for the LMSR could be improved—agents aren't just risk-averse myopic utility maximizers. They exhibit strategic behavior and may try to "game" the market by timing trades.

We addressed this at a high level in Section 4.4. The main issue is that (to the best of our knowledge), no model of strategic behavior exists that would extend to a market on a real-world event where the information structure is unknown and there are hundreds of traders. Furthermore, it is unclear whether a significant amount of traders would attempt this behavior (or be successful at it) in a complex setting like this; previous research has found that when markets do not impose a trading order, strategic behavior is not very predictable. We emphasize that our model shows the *potential* of the LMSR, rather than being an absolute proof that the LMSR is as accurate as the CDA.

2. Agents' belief reports in an actual LMSR market would be different from their belief reports in a CDA market.

This may be true; there is evidence to suggest that some agents misunderstood the task or were careless in reporting beliefs (a clear example being the agent who tried to sell shares at \$0.01 but reported their belief as 99%). We attempted to account for this by eliminating clearly erroneous reports (i.e. those inconsistent with orders by a margin of 0.20 or more). It's entirely possible that with real money at stake in the LMSR, there would be less of these user errors, and we might find that the LMSR becomes even more accurate.

Another argument could be made that agents would not have reported the same belief in an LMSR market because the price evolution would be different. If the CDA price were \$0.05, but the LMSR price simulated from belief reports was \$0.20 at the same point in time, the current trader (assuming they update their priors based on the current price) would hold different beliefs in the LMSR simulation than they reported in the CDA. We elected not to use a model of belief updating in the LMSR simulation, as this would likely involve many more questionable

assumptions than we have made—one example would be the choice of belief updating mechanism agents are assumed to use (since Bayes' rule is clearly intractable here). Do they take a linear combination of their prior and the current price? However, we encourage those who would like to build on this work to consider this as a possible modification.

7.1 Future Work

One direction for future work is to consider other dimensions along which the two mechanisms (CDA and LMSR) differ. There may be other properties strongly correlated with which mechanism is more accurate; for instance, Dana et al. find that Prices are more accurate closer to the market's closing date, while Beliefs are more accurate earlier in the market. By also making the weights a function of the time until the market closes, market operators may achieve even more accurate boosted results.

Furthermore, alternative methods for aggregating the predictions should be considered. We took a linear combination of the two mechanisms while bounding the coefficient between 0 and 1; future work may relax this constraint to allow for the aggregated prediction to be outside the interval determined by the CDA and LMSR predictions. It may also be fruitful to consider other modifications to boost accuracy, such as extremizing predictions or adding a third mechanism to the linear combination. Care must be taken to prevent overfitting, but we suspect that adding more models to the aggregation may be helpful.

In the experimental domain, an interesting experiment might be to run a large number of prediction markets (as the GJP did), but to set up a CDA and an LMSR market for each event. Experimenters could randomize participants to have access to exactly one of the two mechanisms on each event; person A might have access to the CDA market for event 1, the LMSR market for event 2, and the CDA market for event 3. The accuracy of each mechanism could then be compared without worrying about bias introduced by having different events (since each mechanism covers the same events) or different trader characteristics (since the traders are drawn from the same pool and randomized into each mechanism). This would provide an excellent experimental verification of the results obtained in this paper via simulation.

Finally, we think a more in-depth exploration of effect of parameter values would be of great interest. Given the theoretical difficulty of determining the effect of ρ and b on the LMSR mechanism, it seems that the best course of action would be an empirical study in which traders are randomly assigned to LMSR markets with different b values.

8 Conclusion

Overall, our work provides an empirical demonstration that the LMSR mechanism has the potential to be just as accurate as the CDA when running prediction markets, and that this result is relatively robust to the risk attitudes of traders, as well as some choices of the liquidity parameter b. Our work also shows that a belief aggregation mechanism *simulating* the LMSR given belief reports has similar accuracy to the CDA while also being comparable in other metrics like calibration error and discrimination. This provides two contributions to the existing literature: giving the first (to our knowledge) direct comparison of the two mechanisms done in a real-world non-laboratory setting and providing another belief aggregation mechanism with strong performance.

We have also verified the claim that LMSR markets have improved accuracy relative to the CDA in less liquid environments and leveraged this information to create a hybrid model that aggregates the predictions of the two markets in a unique way. In doing so, we have contributed a new mechanism that can augment the accuracy of predictions.

Going forward, people and organizations interested in making accurate predictions may want to spend less time thinking about their choice of mechanism, especially if they plan to run a large number of markets; instead, they may want to consider using multiple mechanisms and aggregating their predictions.

9 References

[1] Berg, J., Nelson, F., & Rietz, T. (2008). Prediction market accuracy in the long run. International Journal of Forecasting, 24(2), 285-300. http://www.forecastingprinciples.com/files/Berg_Nelson_Rietz_2007.pdf

[2] Chen et al. (2007). Bluffing and Strategic Reticence in Prediction Markets. In Internet and Network Economics: Third International Workshop, WINE 2007, San Diego, CA, USA, December 12-14, 2007. http://yiling.seas.harvard.edu/wp-content/uploads/bluffing-long-appendix.pdf

[3] Chen et al. (2009). Gaming Prediction Markets: Equilibrium Strategies with a Market Maker. Algorithmica 58(4): 930-969. https://dash.harvard.edu/bitstream/handle/1/5027877/chen_algorithmica_author.pdf?sequence=1

[4] Chen, Y., & Pennock, D. (2010). Designing Markets For Prediction. AI Magazine, Chen, Yiling and David M. Pennock. 2010. Designing markets for prediction. AI Magazine 31(4): 42-52.

[5] Cowgill, B., & Zitzewitz, E. (2015). Corporate Prediction Markets: Evidence from Google, Ford, and Firm X. The Review of Economic Studies, 82(4), 1309-1341. http://www.restud.com/wp-content/uploads/2015/03/MS14671manuscript.pdf

[6] Dana, J., Atanasov, P., Tetlock, P., Mellers, B. (2019) Are markets more accurate than polls? The surprising informational value of "just asking". Judgment and Decision Making, 14(2), 135-147. http://journal.sjdm.org/18/18919/jdm18919.pdf

[7] Goel, S., Reeves, D., Watts, D., Pennock, D. (2010). Prediction without markets. Proceedings of the 11th ACM Conference on Electronic Commerce, 357-366. https://5harad.com/papers/pred-wo-markets.pdf

[8] Hanson, R. (2002) Logarithmic Market Scoring Rules for Modular Combinatorial Information Aggregation. Journal of Prediction Markets, 1 (1), 1. http://mason.gmu.edu/~rhanson/mktscore.pdf

[9] Healy, P., Linardi, S., Lowery, J., Ledyard, J. (2010) Prediction Markets: Alternative Mechanisms for Complex Environments with Few Traders. Management Science, 56(11), 1977-1996. http://www.its.caltech.edu/~jledyard/John's%20Papers/jl67.pdf

[10] Jian, L., & Sami, R. (2012). Aggregation and Manipulation in Prediction Markets: Effects of Trading Mechanism and Information Distribution. Management Science, 123-140. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.297.1817&rep=rep1&type=pdf

[11] Ledyard, J., Hanson, R., & Ishikida, T. (2009). An experimental test of combinatorial information markets. Journal of Economic Behavior and Organization, 69(2), 182-189.

http://mason.gmu.edu/~rhanson/testcomb.pdf

[12] Parkes, D. and Seuken, S. (2019). Economics and Computation. Cambridge University Press 2019. (draft)

[13] Servan-Schreiber, E., Wolfers, J., Pennock, D., & Galebach, B. (2004). Prediction Markets: Does Money Matter? Electronic Markets, 243-251.

[14] Sethi, R., & Vaughan, J. (2016). Belief Aggregation with Automated Market Makers. Computational Economics, 48(1), 155-178. http://www.columbia.edu/~rs328/hetpriors.pdf

[15] Wolfers, J., & Zitzewitz, E. (2004). Prediction Markets. Journal of Economic Perspectives, 18(2), 107-126. http://www.dartmouth.edu/~ericz/predictionmarkets.pdf

[16] Wolfers, J., & Zitzewitz, E. (2006). Interpreting Prediction Market Prices as Probabilities. NBER Working Paper Series, 12200.