Price Evolution in a Continuous Double Auction Prediction Market With a Scoring-Rule Based Market Maker*

Mithun Chakraborty, Sanmay Das, Justin Peabody
{mithunchakraborty,sanmay}@seas.wustl.edu, peabody@wustl.edu
Department of Computer Science and Engineering, Washington University in St. Louis

Abstract

The logarithmic market scoring rule (LMSR), the most common automated market making rule for prediction markets, is typically studied in the framework of dealer markets, where the market maker takes one side of every transaction. The continuous double auction (CDA) is a much more widely used microstructure for general financial markets in practice. In this paper, we study the properties of CDA prediction markets with zero-intelligence traders in which an LMSR-style market maker participates actively. We extend an existing idea of Robin Hanson for integrating LMSR with limit order books in order to provide a new, self-contained market making algorithm that does not need "special" access to the order book and can participate as another trader. We find that, as expected, the presence of the market maker leads to generally lower bid-ask spreads and higher trader surplus (or price improvement), but, surprisingly, does not necessarily improve price discovery and market efficiency; this latter effect is more pronounced when there is higher variability in trader beliefs.

Introduction

Financial markets, such as those for stocks, bonds, and options, provide participants with opportunities for hedging, investment, and speculation. Prediction markets (which can be thought of as a type of binary option) are often used to forecast future events like elections or sports outcomes, with participants using either real or play money [Berg and Rietz, 2006; Cowgill and Zitzewitz, 2014]. In order to deal with the chicken-and-egg problem of liquidity, many markets employ market makers, specially designated agents that are responsible for providing liquidity by always being ready to transact with traders. In the last decade, research on algorithmic market making has become one of the interesting contact points between artificial intelligence and finance, both in the general context [Das, 2005; 2008; Wah and Wellman, 2014], and specifically in the design of prediction markets ([Hanson, 2003b; Chen, 2007; Brahma et al., 2012; Othman et al., 2013; Abernethy et al., 2014] etc.).

Most research on algorithmic market making in both financial and prediction markets has either focused on market making as a trading strategy [Chakraborty and Kearns, 2011; Schmitz, 2010] or has modeled the market as a pure dealer market, where the market maker takes one side of every trade [Hanson, 2003b; Das, 2008; Othman et al., 2013]. The market can therefore be modeled in terms of the market maker's quoted bid (buy) and ask (sell) prices, and traders' decisions on whether or not to transact at these prices. However, most modern markets, ranging from big financial markets like the NYSE and NASDAQ to smaller prediction markets like the Iowa Electronic Markets, use the continuous double auction (CDA) mechanism [Forsythe et al., 1992]. In CDAs, participants can place *limit orders* that specify a transaction price and are guaranteed to only execute at that price or better (although execution is, of course, no longer guaranteed). The key element of CDAs is the limit order book, which contains all active buy and sell limit orders; the highest buy and the lowest sell constitute the market bid and ask prices at any point in time.

While most practical market making algorithms (for example, those used by market makers on the NYSE and NAS-DAQ) are deployed in markets with limit order books, the academic literature on algorithmic market making has thus far produced almost no analysis of the impact of market making in CDA markets (with the exception of [Wah and Wellman, 2014]). Here we begin to tackle this problem in the context of market making in prediction markets. The logarithmic market scoring rule proposed by Robin Hanson [2003b] is probably the most commonly deployed automated market maker in prediction markets. Hanson [2003a] also provides a scheme for integrating order books with his market making algorithm which, to the best of our knowledge, has not yet been evaluated in the literature. This scheme, as proposed, involves the market maker having special access to orders before they hit the order book, and a "parallel" implementation that looks at the incoming order, the order book, and executes portions of the trade with the

---

market maker and portions with the existing orders on the order book. In addition to the special system privileges this requires, it is not entirely transparent to traders, since the order books themselves never reflect the market maker's presence (and thus give a worse impression of the state of prices and the bid-ask spread than reality).

In this paper, we propose a modification of Hanson's scheme for integrating LMSR with CDA mechanisms that allows an LMSR-based market making agent to compute limit bid and ask prices and participate in the order books as any other trader would, while still maintaining the key desirable properties – namely improved liquidity with bounded worst-case loss. We call this the "Integrated" market maker (as opposed to the "Parallel" market maker of the original scheme). In general, analysis of the properties of market making algorithms in practice is difficult, since they affect the dynamics of the pricing mechanism itself, and therefore the standard practice of backtesting on historical data is of very limited value. However, there is evidence that simulation models with zero-intelligence (ZI) traders [Gode and Sunder, 1993] can replicate many key features of limit order book dynamics [Farmer, Patelli, and Zovko, 2005; Othman, 2008] and have practical value in assessing the properties of market making algorithms [Brahma et al., 2012]. Therefore, we evaluate market properties in prediction markets populated by ZI traders; we compare the parallel and integrated implementations of LMSR with a situation where no market maker is present, and also a pure dealer market mediated by LMSR. We are mainly interested in the following properties:

- Information aggregation properties: For example, how fast does the market price converge to the true underlying asset value? How far away is the price from the true value, on average?

- Market quality properties: For example, how liquid is the market, as measured by the bid-ask spread? How much surplus or price improvement does the market generate?

In our experiments we find that the presence of the market maker leads to generally lower bid-ask spreads and higher trader surplus (or price improvement), but, surprisingly, does not necessarily improve price discovery and market efficiency; this latter effect is more pronounced when there is higher variability in trader beliefs.

Market Model

In this section we describe the precise market model we use and the algorithms used for trading and market making. We simulate four different market microstructures: (1) A Continuous Double Auction (CDA) mechanism without any market maker (pureCDA); (2) A CDA with the "Integrated" implementation of LMSR (INT); (3) (2) A CDA with the "Parallel" implementation of LMSR (PAR); (4) A pure dealer framework with all trades going through a traditional LMSR market maker (pureLMSR).

Prediction market We focus on a prediction market set up to forecast whether a single extraneous uncertain event,

which can be modeled as a binary random variable X, will occur at some pre-determined future date; on that date, the market terminates, and every unit (share) of the asset traded in the market is worth \$1 if the event occurs and is worthless otherwise; we call this cash equivalent of the asset its liquidation value or true value. Before that date, anyone can place orders to buy or (short-)sell any amount of the asset in the market at prices in the interval $[0, 1]$, i.e. the market institution does not impose any budget constraints on traders. We also assume that there is a fixed probability distribution with $\Pr(X = 1) = p_{\text{true}}$ from which the realization of X is drawn on the termination date so that the expected "true" value of the asset is p_{true}, but no agent in the world knows this p_{true} precisely.

Types of orders Traders in a financial exchange can typically place buy/sell orders of two kinds: (1) *market orders* that specify only a quantity and demand immediate execution, hence accept any price offered by the other party, and (2) *limit orders* that specify both a quantity and a limit on acceptable transaction prices (called a limit price or marginal price) but are not guaranteed execution. A Continuous Double Auction (CDA) maintains two *order books*, one for buy orders (bids) and the other for sell orders (asks), which are two priority queues for outstanding limit orders prioritized by limit price and arrival time (higher priority is assigned to a buy order with a higher bid price and a sell order with a lower ask price). Any incoming limit order is placed on the appropriate book, and the mechanism automatically checks to see if the current best (highest) bid is at least as large as the current best (lowest) ask; if yes, then the smaller of the two quantities ordered is traded at the limit price of the order that arrived earlier, the books are updated, and this is continued till the best ask exceeds the best bid. Any new market order is executed immediately, perhaps partially, against the best available outstanding order(s) or is rejected if the book on the other side is empty. In our simulations, all traders place limit orders only but some of them can become market orders effectively, e.g. if an incoming limit buy order "crosses" the books. i.e. its bid is no less than the best ask(s) on the sell order book, and its demand does not exceed the supply of said booked order(s).

Logarithmic Market Scoring Rule We now briefly describe the LMSR market maker for a single-security prediction market liquidating in $\{0, 1\}$ [Hanson, 2003b; Chen, 2007]. Its "state" is described by a real scalar q_{mm}, interpreted as the net outstanding quantity of the security; its instantaneous price at this state, i.e. cost per share of buying/selling an infinitesimal amount from/to LMSR, is given by $p_{\text{mm}} = \frac{e^{q_{\text{mm}}/B}}{1+e^{q_{\text{mm}}/B}}$ where $B > 0$ is a parameter controlling all properties of the market maker. A trader placing a market order for buying any finite quantity Q of assets from LMSR would have to pay it a dollar amount $C(q_{\text{mm}}; Q) = B \ln \left(\frac{1+e^{(q_{\text{mm}}+Q)/B}}{1+e^{q_{\text{mm}}/B}} \right)$ and after the transaction, the market maker's state is updated to $(q_{\text{mm}} + Q)$; for a sell order, the same formula applies by setting Q to the nega-

tive of the supplied quantity, and $-C(q_{mm}; Q) > 0$ becomes the sales proceeds. One key property of LMSR is that it's loss is bounded (for the binary case by $B \ln 2$).

Population of traders Every agent other than the market maker is called a "background" trader [Wah and Wellman, 2014]. Before every simulation, the expected true asset value p_{true} is chosen at random from a common-knowledge common prior which is a uniform distribution on $[0, 1]$. Every trader i then observes a private sequence of N_{trials} Bernoulli trials with probability of success p_{true}, and sets her idiosyncratic valuation of the asset to her Bayesian posterior expectation of the true value, $v_i = \frac{x_i + 1}{N_{trials} + 2}$ where x_i is the number of successes in her sample. Thus, N_{trials} is a measure of the precision of the signal that each trader receives, related to the inverse of the variance of beliefs across the population, similar to the model of Zhang *et al* [2012]. The implementation of a trading decision on top of the belief then follows the zero-intelligence (ZI) trader model [Gode and Sunder, 1993; Othman, 2008], with the addition of non-unit trade sizes. At each step of a simulation (a "trading episode"), a trader is picked uniformly at random and is assigned buyer or seller status with equal probability except for pureLMSR (see below). She then places her limit order, the limit price being drawn uniformly at random from $[v_i, 1]$ if she is a seller and from $[0, v_i]$ if she is a buyer, and the order quantity from a common exponential distribution with mean $\lambda = 20$ which is known to the market mechanism.

(1) pureCDA We have already fully explained the interaction between a CDA mechanism with no market making and the trading population under **Types of Orders**.

(2) PAR The parallel implementation is a single-security version of Robin Hanson's "booked orders for market scoring rules" [Hanson, 2003a]. We delineate its operation for a buy order, the treatment of sell orders being symmetric. Suppose a limit buy order for a quantity q_b at a limit price (bid) p_b arrives when the LMSR market maker's instantaneous price is p_{mm}, and the current best bid and ask prices are b_{max}, a_{min} (at market inception, both books are empty, and $p_{mm} = 0.5$). If $p_b \leq p_{mm}$[1], the order cannot be immediately executed, so it is pushed on to the limit buy order book. If $p_{mm} < p_b$, and q_b is not large enough to drive p_{mm} beyond $\min\{p_b, a_{min}\}$, then the incoming order is completely executed with the market maker according to the traditional LMSR algorithm; otherwise, if $p_{mm} < p_b < a_{min}$, it is only partially executed with LMSR till p_{mm} reaches p_b, the residual order being placed on the buy order book; but if $p_b \geq a_{min}$, LMSR sells only till its instantaneous price hits a_{min} after which the incoming order executes against the best booked ask. If the top level of the book is exhausted but the incoming order is not, LMSR is invoked again, and this process recurs till either the order is finished or the new best ask exceeds the order's bid price. The loss bound of the standard LMSR algorithm is maintained in this case.

---

[1]In this implementation, p_{mm} always lies between the best ask and bid prices on the books, so $p_b \leq p_{mm}$ implies that p_b does not exceed the minimum ask price either.
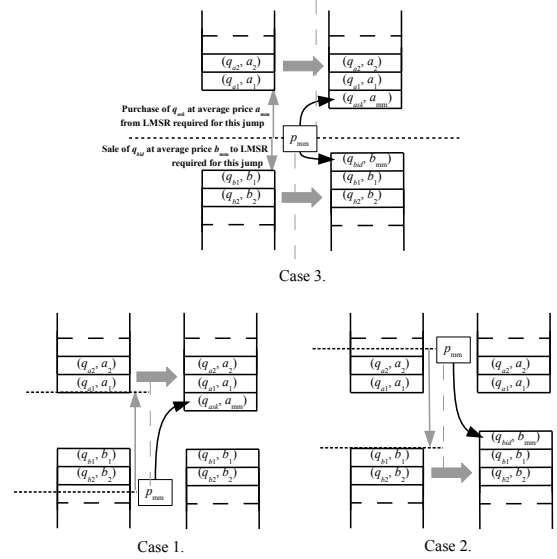

Figure 1: Illustration of how the market maker in the INT setting places ask and bid quotes every time the state of the books changes.

(3) INT In this novel "integrated" implementation that we propose, whenever the best ask and bid prices on the books change, an LMSR-based agent steps in.

1. If its instantaneous price $p_{mm} \leq b_{max}$, then LMSR generates only a limit sell order for a quantity $q_{ask} = B \ln\left(\frac{1/p_{mm} - 1}{1/a_{min} - 1}\right)$ at an ask price of $\frac{B}{q_{ask}} \ln\left(\frac{1 - p_{mm}}{1 - a_{min}}\right)$.

2. If $p_{mm} \geq a_{min}$, then it generates a buy order for $q_{bid} = B \ln\left(\frac{1/b_{max} - 1}{1/p_{mm} - 1}\right)$ at a bid of $\frac{B}{q_{bid}} \ln\left(\frac{1 - b_{max}}{1 - p_{mm}}\right)$.

3. If $b_{max} < p_{mm} < a_{min}$, both orders are generated.

Note that if fully executed immediately these orders would take the LMSR price to b_{max} and a_{min} respectively. The LMSR trader then replaces all its earlier orders with the new order(s) if this action does not immediately cross the books, otherwise it sits idle. After this step, the market is now ready to accept a new order from the background traders, or continue the execution of a partially filled outstanding order, as the case may be. Thus, this market maker can be implemented in practice as just another trader, which is a significant benefit over the PAR framework where the market maker requires some special access to incoming trades and order books. Moreover, any feasible trade with the INT market maker is executed at its actual quoted price rather than following the non-linear LMSR pricing function, which makes trading more transparent and intuitive to traders.

The original LMSR loss bound again holds. Also, we can prove that INT myopically imposes at least as high a cost on the next arriving trader as PAR, assuming that the market makers and order books are in the same state.

Proposition 1. *Suppose the LMSR market maker in both PAR and INT are in state q, and the order books are also*

otherwise identical. For any next arriving trade, the immediate cost incurred by the next trader is at least as high for INT as it is for PAR.

Proof Sketch. Consider the last of the three cases for INT above, $b_{\text{max}} < p_{\text{mm}} < a_{\text{min}}$, and let Q^* be the quantity one would need to buy from LMSR to bring its price to a_{min}. Then, if the current state of the INT market maker is q, it will place a sell order of Q^* at an ask of $\frac{C(q;Q^*)}{Q^*}$. Now if a buy order for $Q < Q^*$ arrives with a sufficiently high bid, the whole of it will execute with the market maker, and the immediate earnings of the latter will be $\frac{QC(q;Q^*)}{Q^*}$. If the PAR market maker had the same state q (hence the same p_{mm}) when the same buy order arrived, the ensuing trade would cost the trader $C(q;Q)$ which is less than INT's earnings since $\frac{C(q;Q)}{Q} < \frac{C(q;Q^*)}{Q^*}$ from the convexity of C. Similar arguments apply to the other cases. $\square$

This result suggests that INT might provide somewhat less liquidity in general than PAR, and incur less loss in doing so, but we do not expect them to be very different. However, this is a loose prediction, since the result is myopic – it says nothing about price evolution in a market; given the market maker's active role, the dynamics of the evolution of q and the order book could conceivably end up quite different. We examine this issue further in the experiments.

(4) pureLMSR In this setting, traders still place limit orders but an LMSR market maker takes one side of *every* trade. At each trading episode, a trader arrives and compares her private valuation v_i to the current market price p_{mm}. If $v_i > p_{\text{mm}}$, she decides to buy; if $v_i < p_{\text{mm}}$, she decides to sell, and leaves without placing any order otherwise. Then she picks her limit price and order size exactly as the ZI traders above. The quantity bought/sold is the minimum of the order size and the quantity needed to drive the LMSR's instantaneous price to the trader's limit price, and monetary transfers are determined by the above function $C(\cdot;\cdot)$.

Note that all components of each limit order of a trader are independent of the market state for all four settings, except for the direction of the trade (buy/sell) in pureLMSR.

Evaluation

We present an overview of the various measures we use to evaluate the properties of our market environments.

Information aggregation properties:

- **ConvTime** (Convergence time): This is defined as the number of trading episodes it takes for the "market price" p_{M} to get within a band of size ± 0.05 around the true expected asset value p_{true} for the first time; $p_{\text{M}}(t)$ is measured at the end of every trading episode t as the mid-point of the bid-ask spread $((b_{\text{max}}(t) + a_{\text{min}}(t))/2)$ for each of the models with CDA, and as the LMSR instantaneous price for the pure dealer case.[2] Thus,

  ConvTime $= \min\{t : p_{\text{M}}(t) \in [p_{\text{true}} - 0.05, p_{\text{true}} + 0.05]\}$.

  A lower convergence time means that the market's esti-

mate (price) quickly gets close to the true expected asset value, i.e. the market is efficient.

- **RMSD** and **RMSD$_{\text{eq}}$**: RMSD is the root-mean-squared deviation of the market price (defined above) from p_{true} over the entire simulation (n_{trades} trading episodes). RMSD$_{\text{eq}}$ is the root-mean-squared deviation between the same quantities but over only the "equilibrium period", i.e. for $t \geq$ ConvTime. Lower values of these measures indicate lower price volatility, another desirable property from an information aggregation perspective.

Market quality properties:

- **Spread** and **Spread$_{\text{eq}}$**: For each scenario with a CDA, the market bid and ask prices $b_{\text{M}}(t)$ and $a_{\text{M}}(t)$ at the end of each trading episode are the highest bid b_{max} and the lowest ask a_{min} on the books respectively (set to 0 and 1 if the corresponding book is empty). For the pure dealer setting, we assume that the market maker knows the average order size λ of the trading population, so for a current market state of q_{mm}, the effective market quotes are taken to be $a_{\text{M}} = \frac{C(q_{\text{mm}};\lambda)}{\lambda}$ and $b_{\text{M}} = \frac{-C(q_{\text{mm}};-\lambda)}{\lambda}$ which are the prices per share of buying and selling λ shares from and to LMSR at the current state respectively. In our notation, "Spread" denotes the bid-ask spread $(a_{\text{M}}(t) - b_{\text{M}}(t))$ averaged over all n_{trades} episodes, while "Spread$_{\text{eq}}$" is the average taken over the equilibrium period only, as above. The bid-ask spread is widely used as a proxy for market liquidity and smaller values are better, since they imply lower trading costs.

- (Idiosyncratic) **TraderSurplus**: If a trader with idiosyncratic valuation v places a buy order of which a quantity q goes through at an execution price p_{exec}, then the trader's surplus is defined as $q(v - p_{\text{exec}})$ (similarly, a seller's surplus is $q(p_{\text{exec}} - v)$). TraderSurplus denotes the sum of individual surpluses of all background traders. Also note that $(v - p_{\text{exec}})$ and $(p_{\text{exec}} - v)$ correspond loosely to the notion of price improvement, when weighted by the probability of execution at that difference. So, even in settings where the private or idiosyncratic value assumption is untenable, the surplus is still a useful measurement of how much value participants are getting from being in one particular microstructure over another. Since every order executes at a price at least as desirable as its limit price, all trader price improvements (surpluses) are positive.

- **MMloss**: This is the loss incurred by the market making mechanism, computed just like (the negative of) a trader surplus, with the private valuation replaced with the true expected asset value p_{true}. Obviously, this does not apply to pureCDA. Since the market is an ex post zero-sum game between the market maker and the trading population, this measure is also numerically equal to the true expectation of the traders' collective net payoff. This measure is particularly important when the market institution itself subsidizes the market maker.

Results

We ran three sets of 1000 simulations each. In each set, we used a different value of the parameter N_{trials} (20, 40, 100)
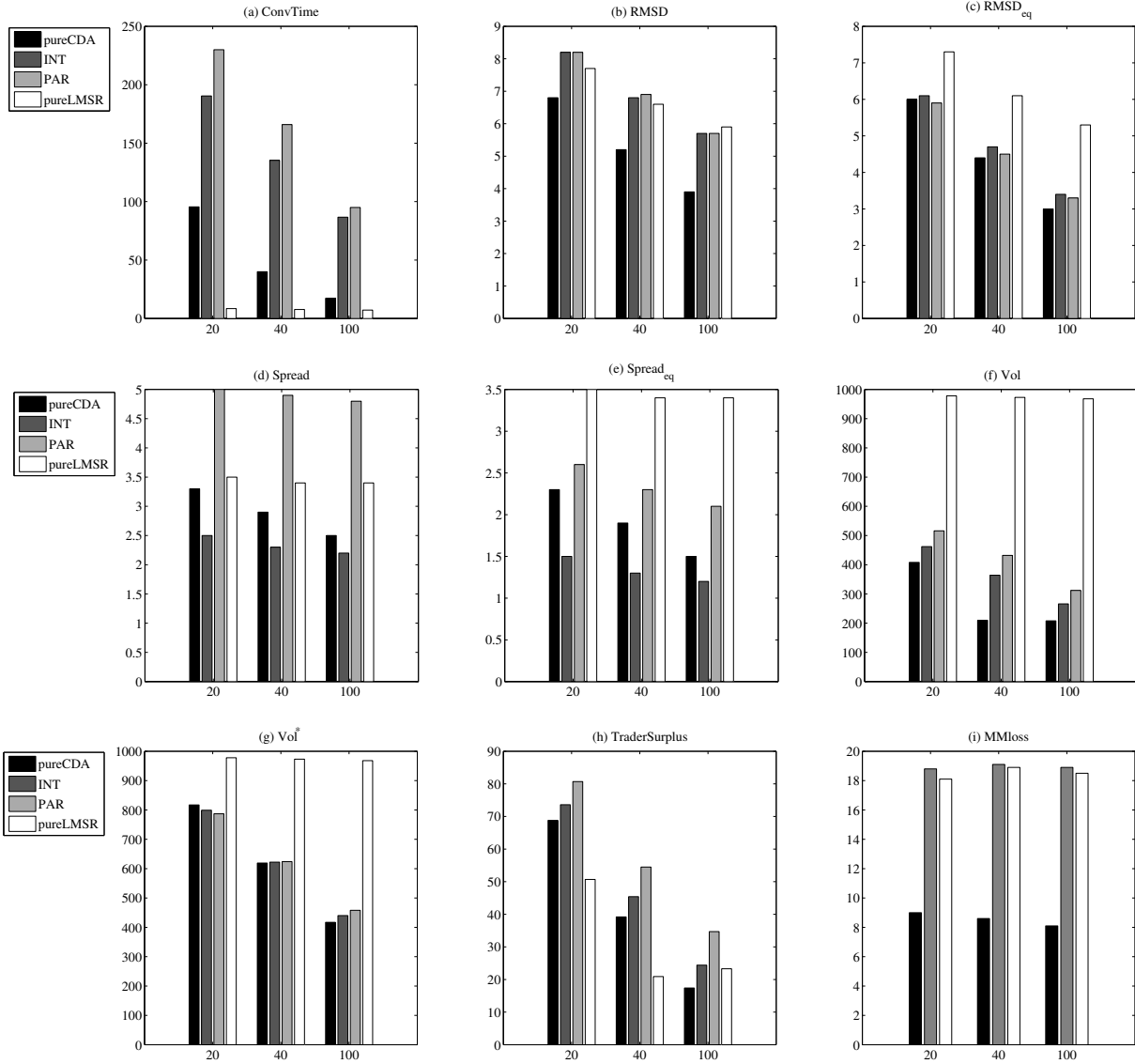
---

[2]If the market price does not enter this band over the duration of the simulation, $ConvTime$ is set to $n_{\text{trades}} = 500$; in our simulations, this is rarely observed.

Figure 2: Experimental results, averaged over 1000 simulations each. The labels along the horizontal axis indicate the number of private Bermoulli trials with success probability p_{true} observed by each trader in the respective simulation set; this number is directly related to the precision in trader beliefs. Values along vertical axis units are in cents in panels (c)-(f) and in dollars in (h), (i).

controlling the precision of trader beliefs. In each simulation, we made the same random sequence of $n_{\text{trades}} = 500$ traders interact with each of our four microstructures. The LMSR parameter B is fixed at 100 for all simulations. We computed all of the above measures for each simulation, and then averaged them over all 1000 simulations. The results are presented in Figure 2, and the analysis follows. Note that, the values (rmsd of prices, spreads) depicted in Figures 1(c)-(f) are in cents while those in the last two figures (surplus, losses) are in dollars, for clarity.

Information aggregation: ConvTime (a) follows the pattern: pureLMSR $<<$ pureCDA $<$ INT $<$ PAR. However, in terms of stability (RMSD, overall (b) and in equilibrium (c)), pureCDA fares the best and the two hybrid mechanisms are very close to each other. The quick convergence and high volatility of LMSR are well-known; surprisingly, coupling it with a CDA delays convergence drastically, but it does ensure more stable prices (lower RMSD_{eq}) once the price converges. While it seems that the market maker-CDA combination might impede the market's learning abilities, it is likely in this case to be an artifact of the fixed beliefs held by ZI traders, who stick to their beliefs no matter what happens to the price – it's not clear that any scoring rule style of market maker would be able to learn quickly when the signals have high variance and the traders don't update their signals. This hypothesis is borne out by the fact that the effect diminishes as the variance in traders' beliefs decreases.

Liquidity / Trading activity: Perhaps the biggest reason to deploy a market-maker is to reduce spreads. Figures 1 (d) and (e) show that INT serves this purpose more effectively than pureCDA. The behavior of PAR, which seems to induce very high spreads, is surprising. This behavior is because we measure the market bid and ask only after the extraneous LMSR agent has intervened and perhaps cleared some orders which would still be waiting in the books in the absence of a market maker, so the spread looks artficially large, compared with pureCDA. In addition, PAR doesn't actually place any new orders on the books, since it waits for orders to arrive before acting, as opposed to INT, which proactively improves spreads by adding to the order book. This finding, which casts doubt on the meaningfulness of spread measurement for PAR, is problematic since many real-life traders use the spread to gauge market quality and make decisions.

To get a better idea of the market maker's role in improving trading activity, we also computed the actual volume of trade executed. We did this in two ways: for each simulation, we maintained a ledger where each entry recorded the buyer, seller, execution price, and quantity of every market trade; after n_{trades} episodes, we added all these traded quantities together to obtain Vol=quantity absorbed by buyers and market maker (if present)=quantity supplied by sellers and market maker. PAR beats both pureCDA and INT with respect to this measure.

We also calculated an alternative measure of trading volume by subtracting the total residual quantity on the order books at the end of each simulation from the total quantity ordered by all traders: Vol* = quantity absorbed by buyers (from sellers and market maker) + quantity supplied by sellers (to buyers and market maker). It double-counts,

perhaps appropriately, every quantity traded between background traders, and thus reflects the overall "satisfaction" of the entire background trader population in a way that the previous measure does not. [3] Strangely, for higher variability in trader beliefs, PAR gives the worst Vol* bettered by INT and pureCDA, but there is a complete reversal in this behavior as the variability decreases. Based on observations of some sample trade ledgers and order book residuals, we believe that the reason is this: in any CDA with a market maker, the market maker gets the advantage of immediacy due to its continuous presence and itself undercuts some of the background traders, thereby reducing the (double-counted) quantity that changes hands between these traders. Hence pureCDA, where every trade must occur between background traders, has a higher Vol*. But with increasing N_{trials} as trader beliefs get closer to each other, relatively more traders trade with other background traders, who now offer competitive prices themselves. This is an interesting example of how the presence of the market maker can affect the dynamics of trade in surprising ways.

Also note that regardless of the microstructure, both Vol and Vol* decrease as the knowledge of the trading crowd gets more and more precise, which is consistent with the idea that as the noise in the beliefs of traders with a common knowledge structure reduces, trading becomes less profitable, hence less likely.

Welfare: Trader surplus or (weighted) price improvement decreases with increasing precision in beliefs but the presence of a market maker consistently improves the surplus as opposed to having only a CDA, PAR more so than INT. It is also noteworthy that the combination of CDA and market making performs better in this respect than each of them individually. Moreover, we consistently observe INT loss $<$ PAR loss $\approx$ pureLMSR loss, and these losses respect the known LMSR loss bound of 69.3 (starting price $= 0.5$, $B = 100$). This empirical observation supports the notion that Proposition 1 (which shows that myopic costs faced by the market maker are lower for INT than for PAR when they start from the same state in terms of q and the order books) might generalize to expected losses over *sequences* of trades from a particular starting point, an interesting direction for theoretical work on the topic (in a handful of our individual simulations, INT made slightly more loss than PAR, which shows that the sequence result cannot hold deterministically).

Discussion

We have introduced a new LMSR-based market making algorithm that applies to a CDA setting, and have compared its properties with three other market microstructures in simulations with basic trading agents. Future research directions include analyzing these market settings with more sophisticated trader models.

---

[3] Of course, for pureLMSR, Vol* =Vol since the market maker takes one side of every trade.

References

Abernethy, J.; Kutty, S.; Lahaie, S.; and Sami, R. 2014. Information aggregation in exponential family markets. In *Proceedings of the 15th ACM Conference on Economics and Computation*, 395–412.

Berg, J. E., and Rietz, T. A. 2006. The Iowa Electronic Markets: Stylized facts and open issues. *Information Markets: A New Way of Making Decisions, edited by Robert W. Hahn and Paul C. Tetlock. AEI-Brookings Joint Center for Regulatory Studies* 142–169.

Brahma, A.; Chakraborty, M.; Das, S.; Lavoie, A.; and Magdon-Ismail, M. 2012. A Bayesian market maker. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, 215–232.

Chakraborty, T., and Kearns, M. 2011. Market making and mean reversion. In *Proceedings of the 12th ACM conference on Electronic commerce*, 307–314.

Chen, Y. 2007. A utility framework for bounded-loss market makers. In *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence*.

Cowgill, B., and Zitzewitz, E. 2014. Corporate prediction markets: Evidence from Google, Ford, and Firm X. In *Proceedings of the 15th ACM Conference on Economics and Computation*, 525–525.

Das, S. 2005. A learning market-maker in the Glosten–Milgrom model. *Quantitative Finance* 5(2):169–180.

Das, S. 2008. The effects of market-making on price dynamics. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*, 887–894.

Farmer, J. D.; Patelli, P.; and Zovko, I. I. 2005. The predictive power of zero intelligence in financial markets. *Proceedings of the National Academy of Sciences of the United States of America* 102(6):2254–2259.

Forsythe, R.; Nelson, F.; Neumann, G. R.; and Wright, J. 1992. Anatomy of an experimental political stock market. *The American Economic Review* 1142–1161.

Gode, D. K., and Sunder, S. 1993. Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. *Journal of Political Economy* 119–137.

Hanson, R. 2003a. Book orders for market scoring rules. *George Manson University*.

Hanson, R. 2003b. Combinatorial information market design. *Information Systems Frontiers* 5(1):107–119.

Othman, A.; Pennock, D. M.; Reeves, D. M.; and Sandholm, T. 2013. A practical liquidity-sensitive automated market maker. *ACM Transactions on Economics and Computation* 1(3):14.

Othman, A. 2008. Zero-intelligence agents in prediction markets. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*, 879–886.

Schmitz, J. 2010. Algorithmic trading in the Iowa Electronic Markets. *Algorithmic Finance (2012)* 1(2):157–181.

Wah, E., and Wellman, M. P. 2014. Welfare effects of market making in continuous double auctions (preliminary report). *Workshop on Agent-Mediated Electronic Commerce and Trading Agent Design and Analysis, AAMAS*.

Zhang, H.; Horvitz, E.; Chen, Y.; and Parkes, D. C. 2012. Task routing for prediction tasks. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, 889–896.