Designing Automated Market Makers for Combinatorial Securities: A Geometric Viewpoint

Prommy Sultana Hossain Xintong Wang Fang-Yi Yu

Summary

By leveraging range query and range update problems (RQRU) in **computational geometry**, we propose a unified framework for designing automated market makers for combinatorial securities. Our key contributions include:

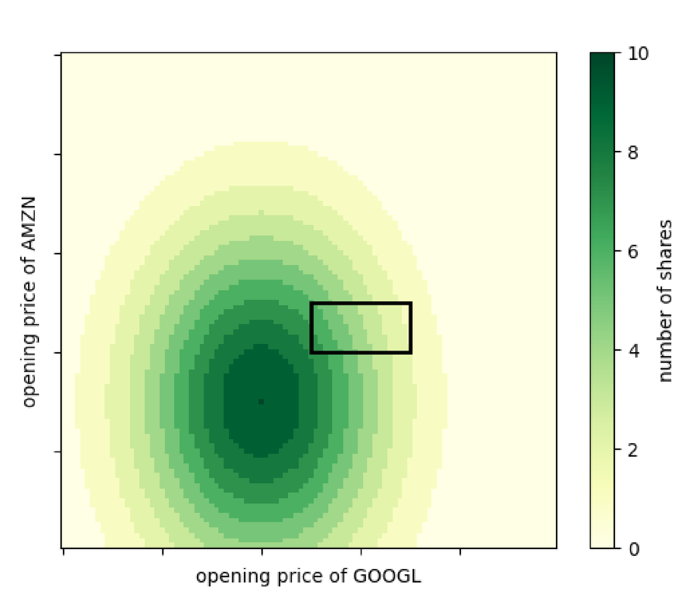
- Designing a partition-tree scheme that supports sublinear time price, cost, and buy query for LMSR when the **VC dimension** is bounded.
- Generalizing to **quadratic scoring rule and $3/2$-power scoring rule** by establishing connections to variants of RQRU.
- Integrating **multi-resolution market designs** into partition-tree scheme.
- Introducing the **combinatorial swap operation problem** in decentralized finance and demonstrating that it can be efficiently reduced to range update problems.

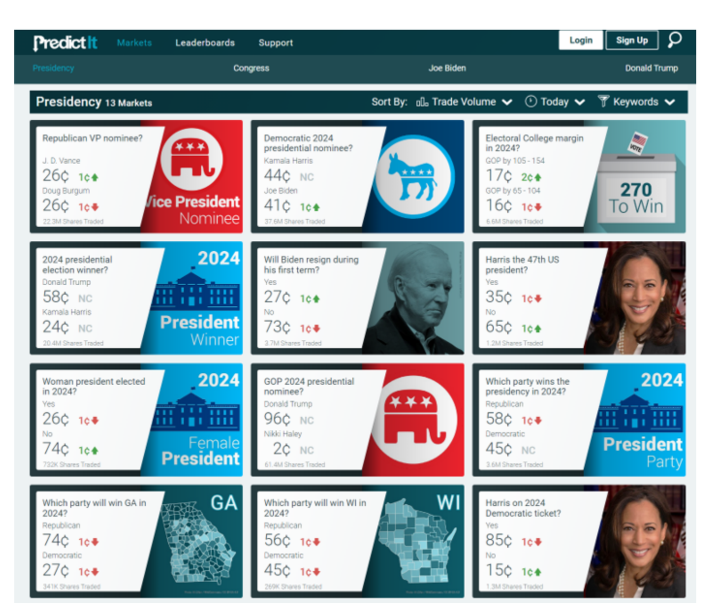
Problem Formulation

Prediction markets with combinatorial securities

We are interested in aggregating predictions on a random variable on $\mathcal{X}$, e.g., the opening value of AMZN and GOOGL at 4pm tomorrow by offering binary securities on a collection of events $E \in \mathcal{F}$.

- Pay \$1 if event E occurs and \$0 otherwise.
- Adjust prices of these securities to elicit predictions.
- For example, $\mathcal{X} = \mathbb{R}^2$ and $\mathcal{F}$ consists of 2D Intervals (e.g., $[180, 220] \times [180, 200]$).





LMSR for combinatorial securities

We consider using logarithmic market scoring rule (LMSR) to adjust price of each securities, $C(\boldsymbol{w}) = b \ln \left(\sum_{x\in\mathcal{X}} e^{w_x/b}\right)$, and offer operations for all $E \in \mathcal{F}$:

- `price`$(E; \boldsymbol{w})$: return the current price of security for E,

$$\texttt{price}(E;\boldsymbol{w}) = \frac{\sum_{x\in E} e^{w_x/b}}{\sum_{x'\in\mathcal{X}} e^{w_{x'}/b}} = \sum_{x\in E}\frac{\partial}{\partial w_x}C(\boldsymbol{w}). \quad (1)$$

- `cost`$(E, s; \boldsymbol{w})$: return the current cost of s shares of security for E,

$$\texttt{cost}(E, s; \boldsymbol{w}) = C(\boldsymbol{w} + s\mathbf{1}_E) - C(\boldsymbol{w}). \quad (2)$$

- `buy`$(E, s; \boldsymbol{w})$: update the state $\boldsymbol{w} \leftarrow \boldsymbol{w} + s\mathbf{1}_E$.

Range Query and Range Update (RQRU)

Range query is a classical problem in computational geometry with many variants. For example, one common objective is to count the number of points or sum of weights within a region E.

$(+, \cdot)$-RQRU

Gives a set system $(\mathcal{X}, \mathcal{F})$, and initial weights $W_0 : \mathcal{X} \to \mathbb{R}_+$. $(+, \cdot)$-RQRU requests a sequence of operations, taking one of the following forms: for any $E \in \mathcal{F}$ and $S \in \mathbb{R}_+$:

- `query`$(E; W)$: compute the total weight of range E, $\sum_{x\in E} W(x)$.
- `update`$(E, S; W)$: update $W(x) \leftarrow S \cdot W(x)$ if $x \in E$, and $W(x) \leftarrow W(x)$ otherwise.

Equivalence

LMSR (market operations)		RQRU (arithmetic operations)
Price and Cost	$\xleftrightarrow{W(x)=e^{w_x/b}}$	Query $(+)$
Buy		Update $(\cdot)$

Applications: Algorithms and Hardness Results

Partition-tree-based scheme

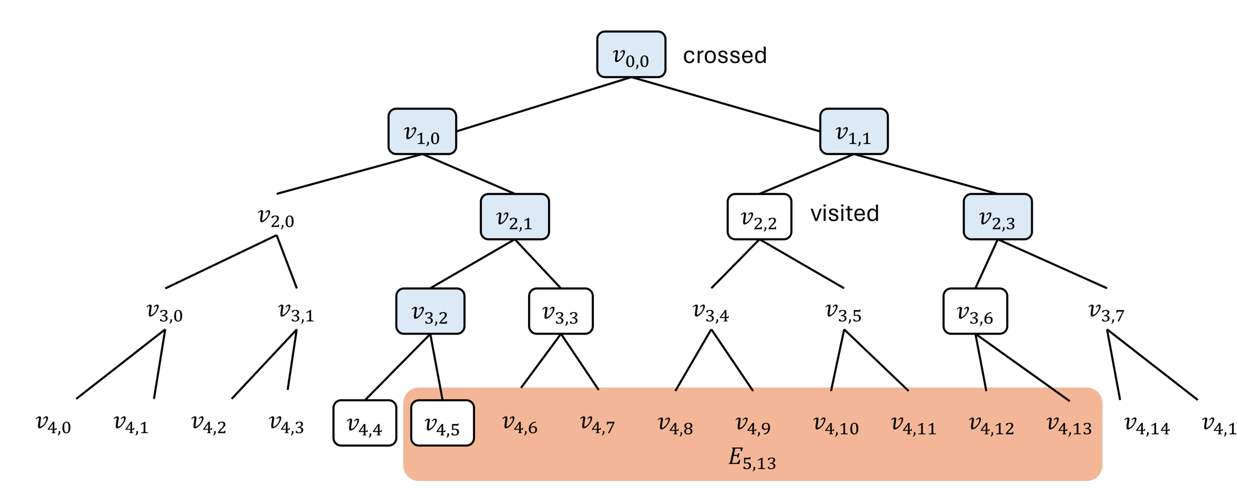


Figure 1. A partition tree for 1D intervals with $n = 16$. The squared nodes are visited by the query and blue ones has node set crossed by $E_{(5,13)}$.

1. 1D intervals: $O(\log n)$, where $n = |\mathcal{X}|$
2. Intervals, and hyperplane in $\mathbb{R}^d$: $O(n^{1-1/d})$
3. Bounded VC: $O(n^{1-\epsilon})$ with $\epsilon > 0$

Hardness results

Introduce several hardness results from computational geometry.

1. 1D intervals: $\Omega(\log n)$
2. 2D intervals: $\Omega(n^\omega)$, connections to solving matrix multiplication
3. Unbounded VC dimension: no $o(n)$ including 1-junta or comparison securities.

Prediction Markets Beyond LMSR

More cost-function-based market making

Table 1. Summary of reductions for AMMs to various range query range update problems

Automated market maker	Query	Update
Logarithmic market scoring rule	addition $+$	multiplication $\cdot$
Quadratic market scoring rule	addition $+$	addition $+$
γ-power market scoring rule	addition $+$	group action

Partition tree and multi-resolution markets

- Example: Predict the opening day of the Gates and Hillman Centers at CMU. We are interested in using different scoring rules for 1) *quarter submarket* (trading securities to bet on during which quarter the center will open), 2) *month submarket*, 3) *week submarket*, and 4) *day submarket*, to facilitate aggregating information at different granularity.
- Show that the multi-resolution design has the same complexity including arbitrage removal given *efficient and local weight updates*.

Decentralized Exchanges

Constant function market maker

Given a finite set of n assets (cryptocurrencies) $\mathcal{X}$, CFMM maintains a *reserve* of available assets $\boldsymbol{w} \in \mathbb{R}^{\mathcal{X}}$ and uses a *trading function* $\varphi : \mathbb{R}^{\mathcal{X}} \to \mathbb{R}$.

- Logarithmic trading function $\varphi(\boldsymbol{w}) = -\sum_{x\in\mathcal{X}} e^{-w_x/b}$
- Constant sum trading function $\varphi(\boldsymbol{w}) = \sum_{x\in\mathcal{X}} w_x$
- Geometric mean trading function $\varphi(\boldsymbol{w}) = \prod_{x\in\mathcal{X}} w_x$

Traders propose to exchange one basket of assets $\boldsymbol{r}^+$ for another $\boldsymbol{r}^- \in \mathbb{R}^{\mathcal{X}}$. The exchange will accept the proposed trade if $\varphi(\boldsymbol{w} + \boldsymbol{r}^+ - \boldsymbol{r}^-) = \varphi(\boldsymbol{w})$ and update the reserve to $\boldsymbol{w} \leftarrow \boldsymbol{w} + \boldsymbol{r}^+ - \boldsymbol{r}^-$.

Combinatorial swap problem

Given $(\mathcal{X}, \mathcal{F})$, a combinatorial swap market maker with φ takes any $E^+, E^- \in \mathcal{F}$ and supports the following operations

- `forward_trade`$(E^-, E^+, s_+, \boldsymbol{w})$: return s so that $\varphi(\boldsymbol{w} + s_+\mathbf{1}_{E^+} - s\mathbf{1}_{E^-}) = \varphi(\boldsymbol{w})$ and update $\boldsymbol{w} \leftarrow \boldsymbol{w} + s_+\mathbf{1}_{E^+} - s\mathbf{1}_{E^-}$.
- `backward_trade`$(E^-, E^+, s_-, \boldsymbol{w})$: return s so that $\varphi(\boldsymbol{w} + s\mathbf{1}_{E^+} - s_-\mathbf{1}_{E^-}) = \varphi(\boldsymbol{w})$ and update $\boldsymbol{w} \leftarrow \boldsymbol{w} + s\mathbf{1}_{E^+} - s_-\mathbf{1}_{E^-}$.

Automated market maker	Query	Update
Log CFMM	addition $+$	multiplication $\cdot$
Linear CFMM	addition $+$	addition $+$
Geometric mean CFMM	multiplication $\cdot$	addition $+$

xintong.wang@rutgers.edu

fangyiyu@gmu.edu