Log-time Prediction Markets for Interval Securities

Xintong Wang *Postdoc@Harvard*

EC Mentoring Workshop, Talk Dissection, July 2022

Miro Dudík
Microsoft Research, NYC

Dave Pennock
Rutgers University

David Rothschild
Microsoft Research, NYC

Prediction Markets for Interval Securities

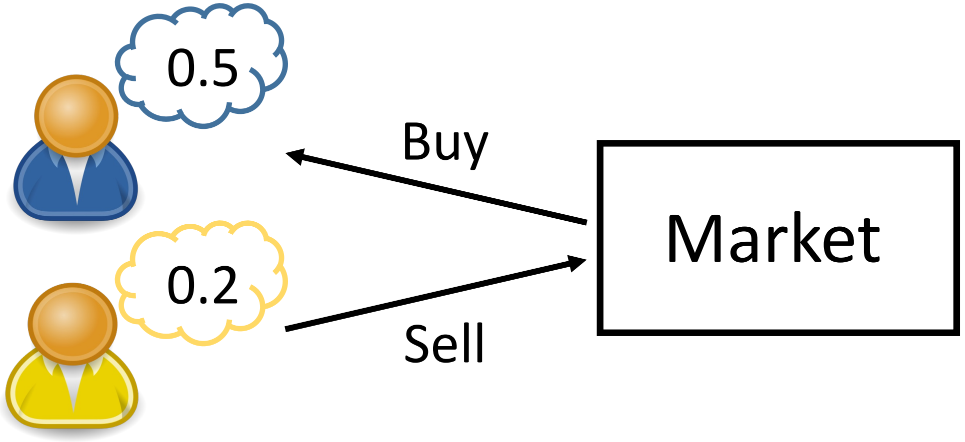
- Prediction Markets
  - Offer securities whose payoff is tied to outcomes of an event.

  E.g., *"the daily commercial air traffic will rise back above 100,000 flights before July 2022".*
  - Traders buy the security for some price, e.g., *\$0.32 per share*.
  - One receives \$1 if true and \$0 if false.

Prediction Markets for Interval Securities

- Prediction Markets
  - Offer securities whose payoff is tied to outcomes of an event.

  E.g., *"the daily commercial air traffic will rise back above 100,000 flights before July 2022".*
  - Traders buy the security for some price, e.g., *$0.32 per share*.
  - One receives $1 if true and $0 if false.



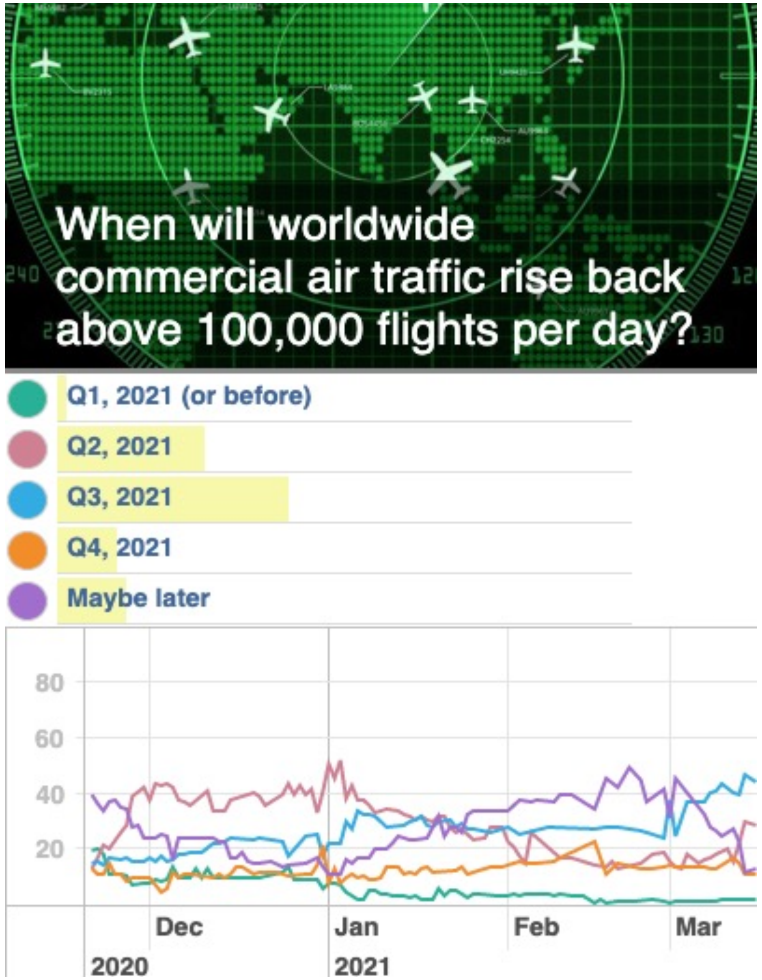
  - Market price reflects a consensus forecast for the event.

Prediction Markets for Interval Securities

- Interval Securities: the outcome will fall into some specified interval.
  - A natural way to elicit prediction about a continuous outcome.

Current Market Implementation

- Require predefined discretization.
- Treat as independent markets.

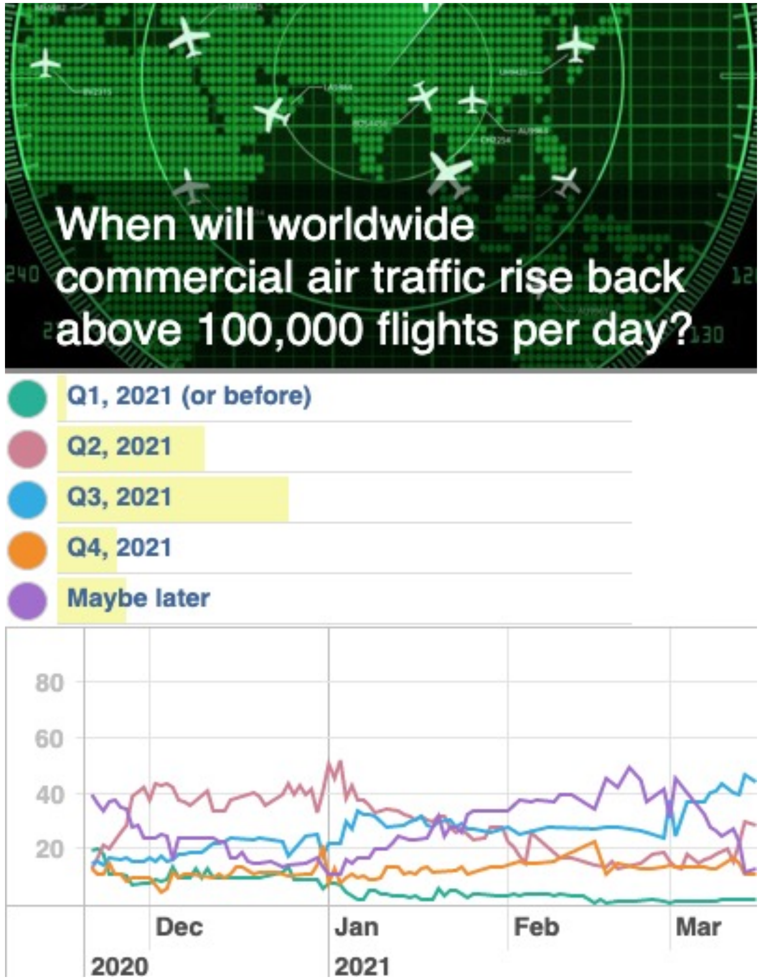


Current Market Implementation

- Require predefined discretization.
- Treat as independent markets.

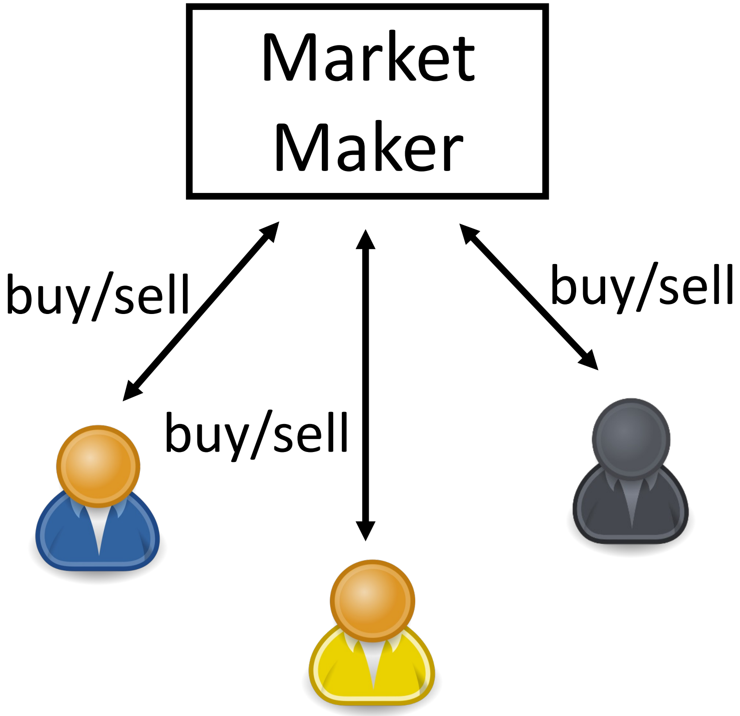
Why not use finer discretization?

Challenge: the thin market problem.



Market Implementation: Automated Market Maker

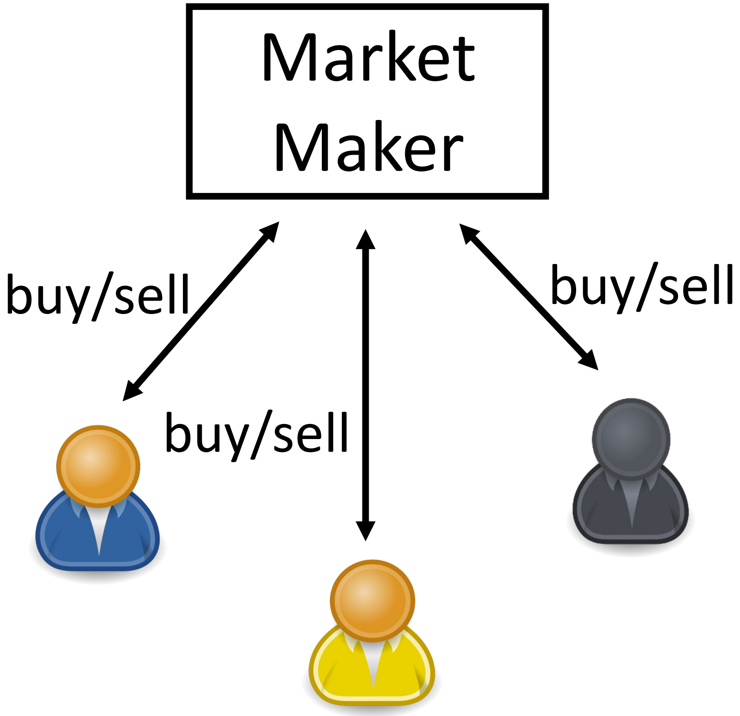
- Set prices and offer to buy or sell *any* interval security at some price.
- If more shares are bought, increase the price of securities on the outcome.
  → reflect a consensus forecast.
- Subsidize the market for information.



Market Implementation: Automated Market Maker

- Set prices and offer to buy or sell *any* interval security at some price.
- If more shares are bought, increase the price of securities on the outcome.
  → reflect a consensus forecast.
- Subsidize the market for information.
- *Challenge: market operations require time linear in the number of outcomes.*

  E.g., quarter (2 bits of precision): runtime 2^2.
  week (6 bits of precision): runtime 2^6.
  day (9 bits of precision): runtime 2^9.



Contribution Summary

The largest amount that the MM has to pay traders across all possible trading sequences and outcomes.

	Market Maker (MM)	Data Structure	Runtime of Market Operations	Worst-Case Loss for MM
previous work	Logarithmic market scoring rule (LMSR) [Hanson 2003]	array	$O(N)$ N = # distinct outcomes	$\log(N)$

Contribution Summary

The largest amount that the MM has to pay traders across all possible trading sequences and outcomes.

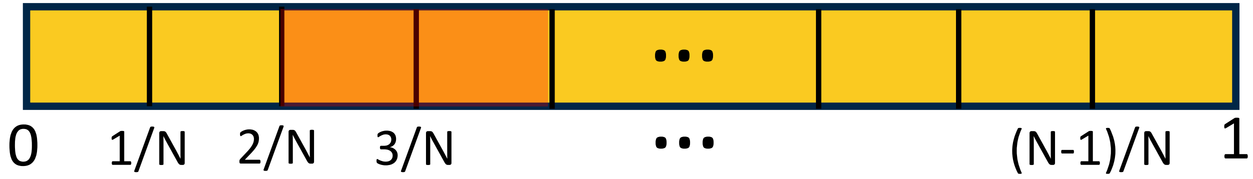
	Market Maker (MM)	Data Structure	Runtime of Market Operations	Worst-Case Loss for MM
previous work	Logarithmic market scoring rule (LMSR) [Hanson 2003]	array	$O(N)$ N = # distinct outcomes	$\log(N)$
this work	Log-time LMSR MM			
	Multi-resolution linearly constrained MM (LCMM)			

LMSR Market Maker - Intuition

LMSR Market Maker - Intuition

- ***price***(I)
  - Keep track of price for each outcome $\omega \in \Omega$.
  - Sum up the prices of all outcomes in the interval, i.e., ***price***$(I) = \sum_{\omega \in I}$ ***price***(ω).
- ***buy***(I, s)
  - Increase the prices of outcomes $\omega \in I$ by a factor of $e^{s/b}$.
  - Renormalize across all prices: prices of bought outcomes↑, prices of others↓.
- *Challenge: **price**(I) and **buy**(I, s) take time linear in the number of outcomes.*

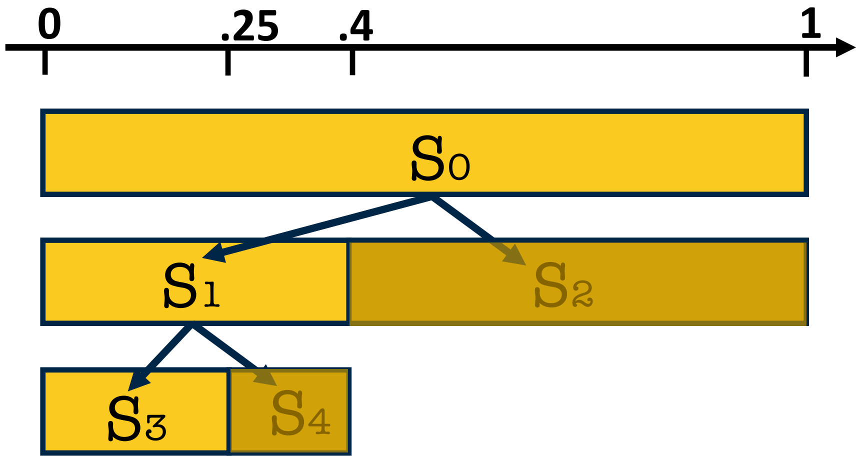
Liquidity parameter set by the market designer.



Outcome Space Ω

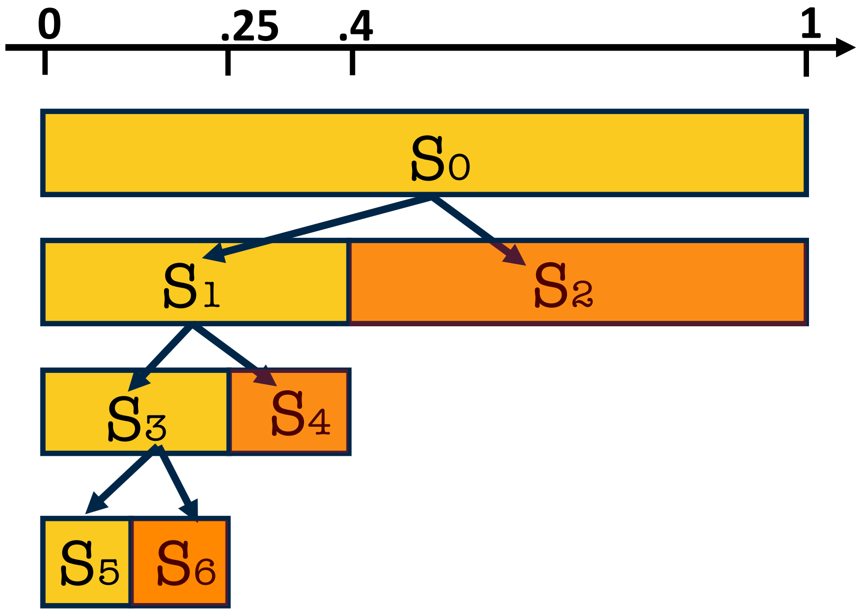
Contribution: Log-time LMSR Market Maker

- A balanced binary tree
  - Construct nodes from queried intervals.
  - Decompose LMSR computations along the tree nodes.
  - Keep track of unnormalized prices (in each node) and partial sums (in parent nodes).
- ***price***(I), e.g., $I = [.25, 1)$
  - Sum up the prices of relevant subintervals (at most $\log n$) along the search path.
  - Normalize by the overall sum (in the root).



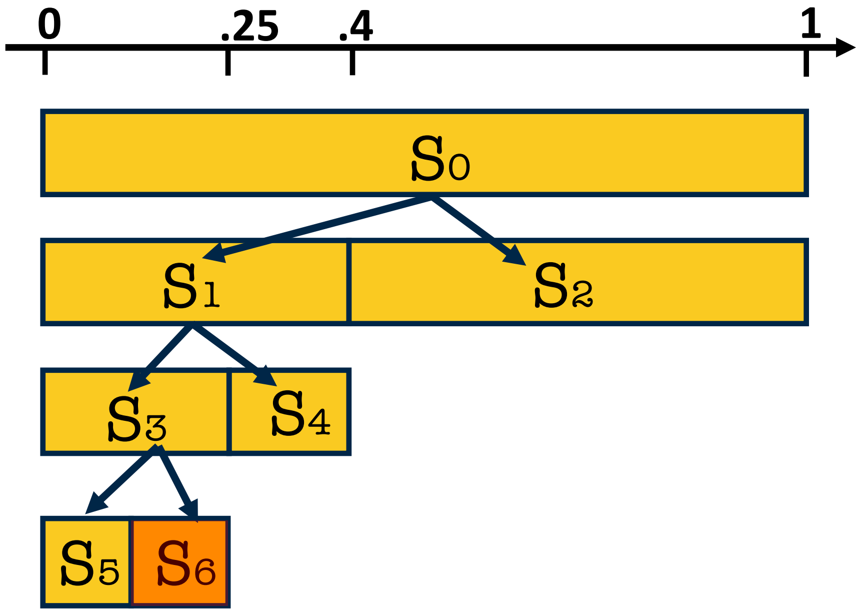
Contribution: Log-time LMSR Market Maker

- ***buy***(I, s)
  - Update the corresponding multipliers of subintervals by $e^{s/b}$ along the search path.
  - Update the partial sums back up.
  - *Challenge: the tree may no longer be balanced!*



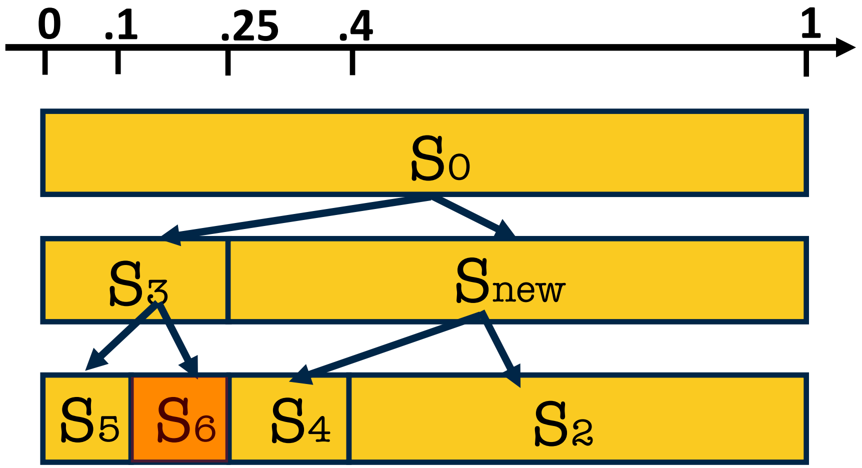
Contribution: Log-time LMSR Market Maker

- ***buy***(I, s)
  - Update the corresponding multipliers of subintervals by $e^{s/b}$ along the search path.
  - Update the partial sums back up.
  - *Challenge: the tree may no longer be balanced!*
  - Rely on rotation to rebalance which requires constant time.



Contribution: Log-time LMSR Market Maker

- ***buy***(I, s)
  - Update the corresponding multipliers of subintervals by $e^{s/b}$ along the search path.
  - Update the partial sums back up.
  - *Challenge: the tree may no longer be balanced!*
  - Rely on rotation to rebalance which requires constant time.



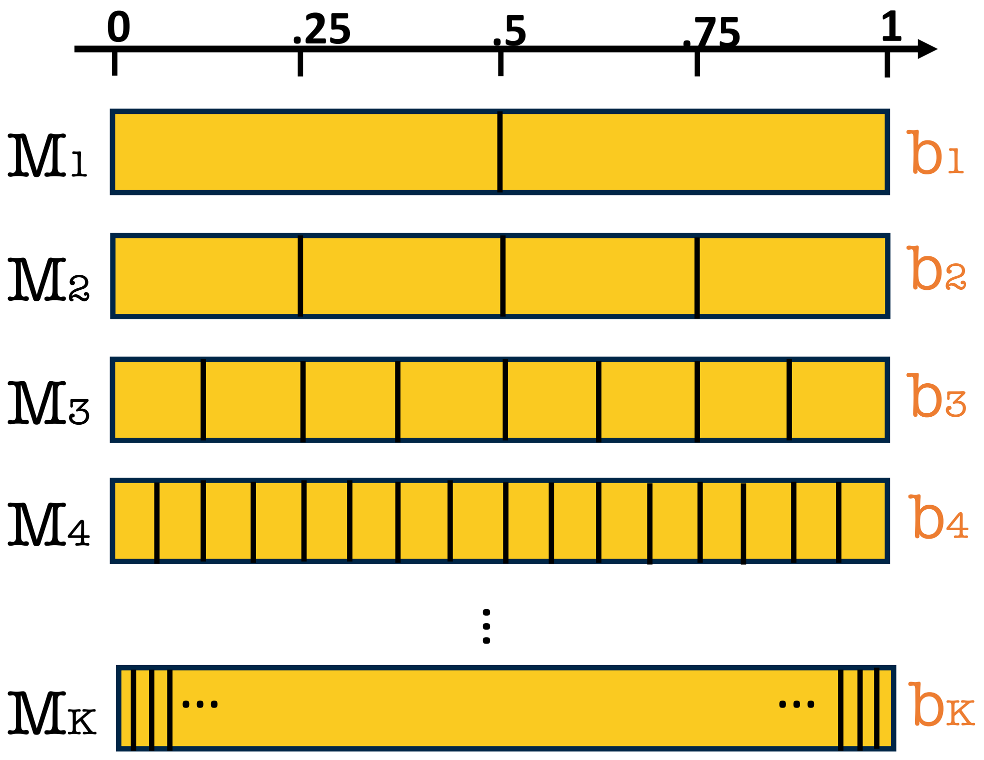
Contribution: Log-time LMSR Market Maker

	Market Maker (MM)	Data Structure	Runtime of Market Operations	Worst-Case Loss for MM
previous work	Logarithmic market scoring rule (LMSR) [Hanson 2003]	array	$O(N)$ N = # distinct outcomes	$\log(N)$
this work	Log-time LMSR MM	binary tree (adaptive)	$O(\log \mathrm{n}) \leq O(\log N)$ n = # distinct queries	$\log(N)$

- *Challenge: worst-case loss is dependent on the number of outcomes.*

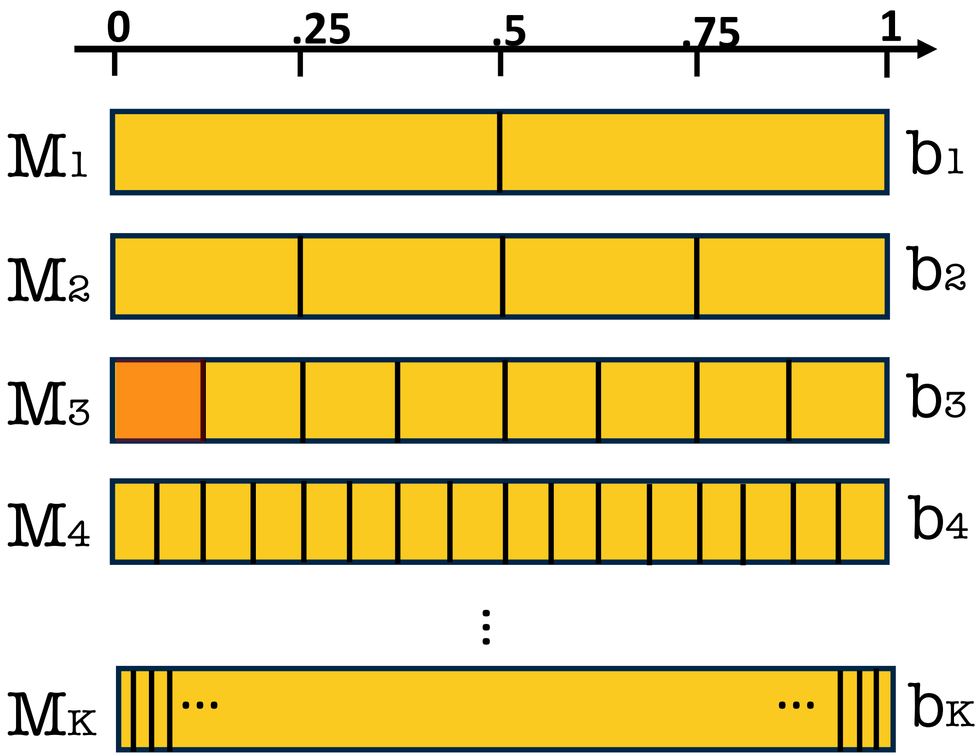
Contribution: Multi-resolution Linearly Constrained Market Maker

- Use multiple LMSRs with different *liquidity parameters* to mediate markets offering interval securities at different resolutions.
- The liquidity parameter controls
  - How fast the price moves, i.e., $e^{s/b}$;
  - The worst-case loss for MM, i.e., $b \log N$.
- Achieve constant loss bound by choosing proper liquidity values.
  - Total worst-case loss:
  $\sum_{k=1}^{K} b_k \log N_k = \sum_{k=1}^{K} b_k \log(2^k)$.
  - E.g., $b_k = O(k^{-2.01})$.



Contribution: Multi-resolution Linearly Constrained Market Maker

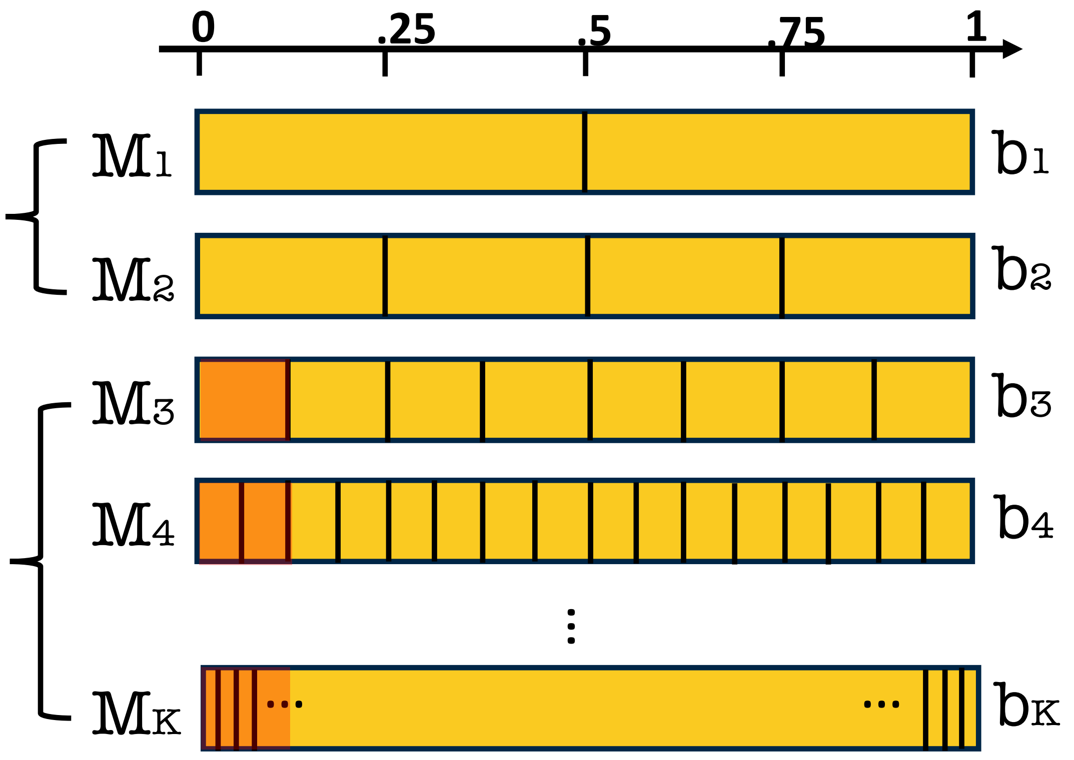
- *Challenge: keep prices coherent across different markets.*
- ***buy***(I, s)
  - Example: ***buy***$(I = [0, .125), 1)$ in M_3 → price incoherence between M_3 and other markets.



Contribution: Multi-resolution Linearly Constrained Market Maker

- The LCMM can remove price incoherence (arbitrage) efficiently across markets.

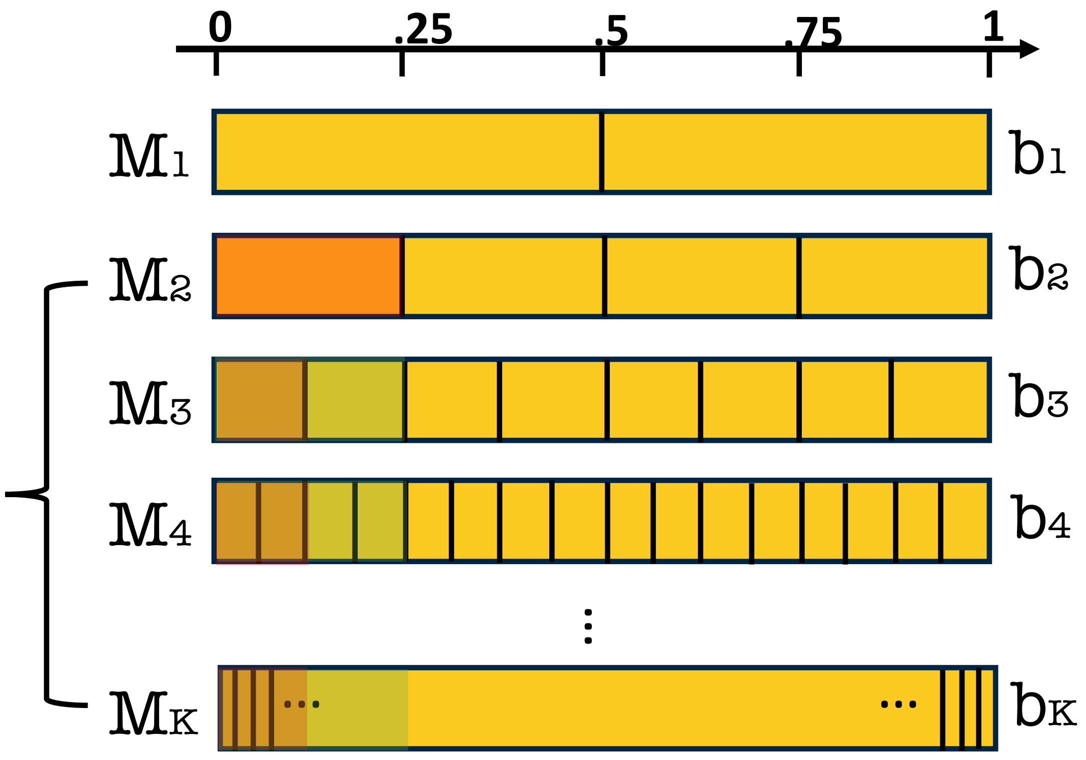
Intuition: split the 1 share among $M_3 \ldots M_k$ according to liquidity ratio to maintain price coherence.



Contribution: Multi-resolution Linearly Constrained Market Maker

- The LCMM can remove price incoherence (arbitrage) efficiently across markets.

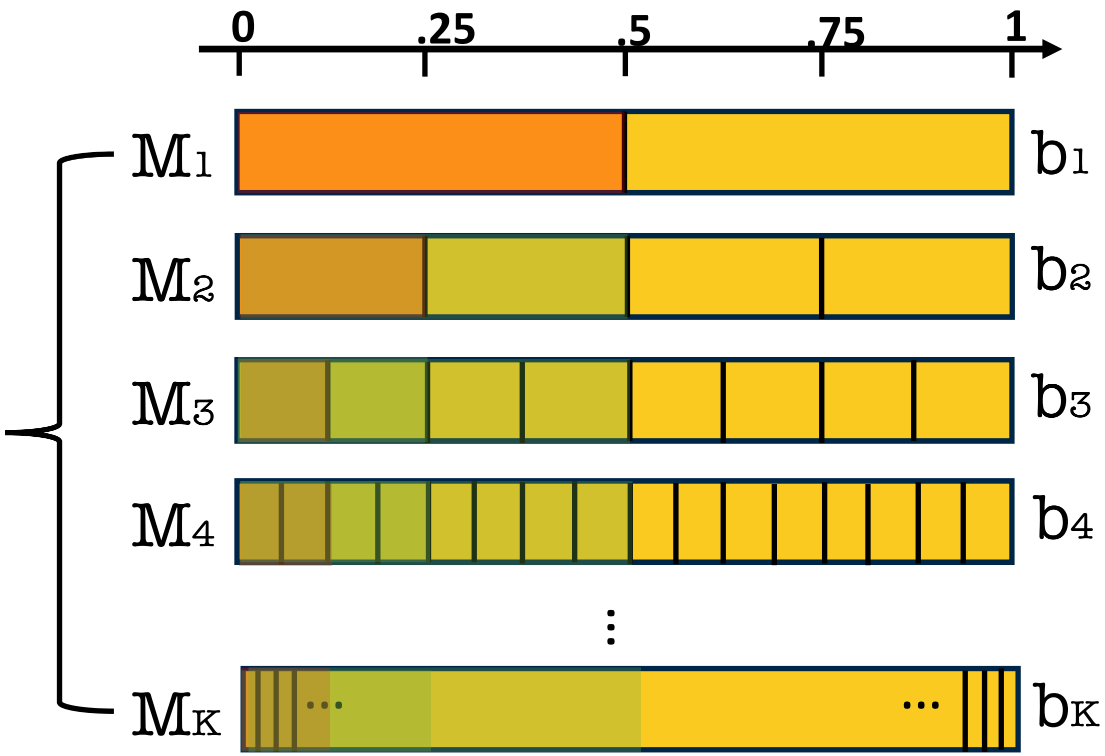
Intuition: remove arbitrage level by level



Contribution: Multi-resolution Linearly Constrained Market Maker

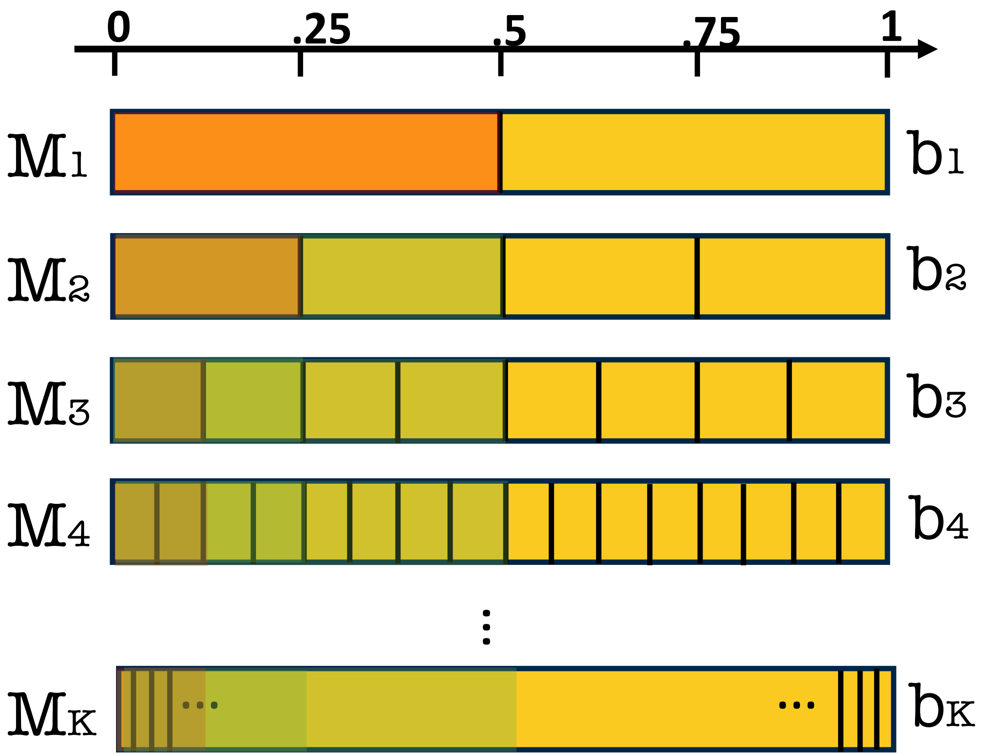
- The LCMM can remove price incoherence (arbitrage) efficiently across markets.

Intuition: remove arbitrage level by level (e.g., buy s' share [0,.5) in M_1 and split sell s' share among M_2… M_k).



Contribution: Multi-resolution Linearly Constrained Market Maker

- The LCMM can remove price incoherence (arbitrage) efficiently across markets.
- A single static binary tree
  - Keep track of (1) trader purchases and (2) automatic purchases made by the LCMM for price coherent.

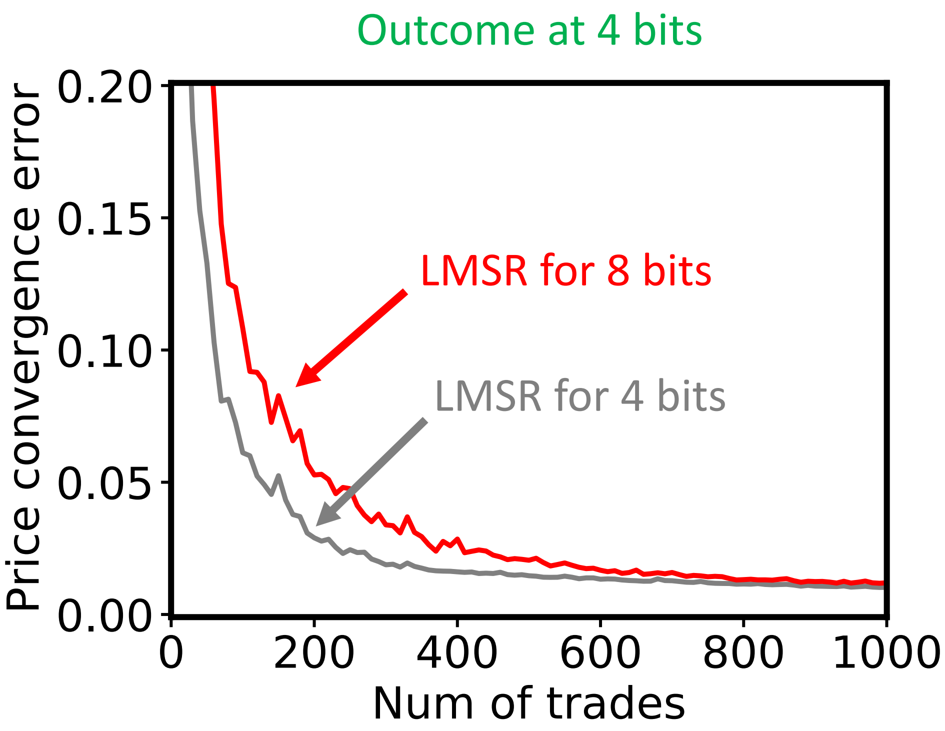


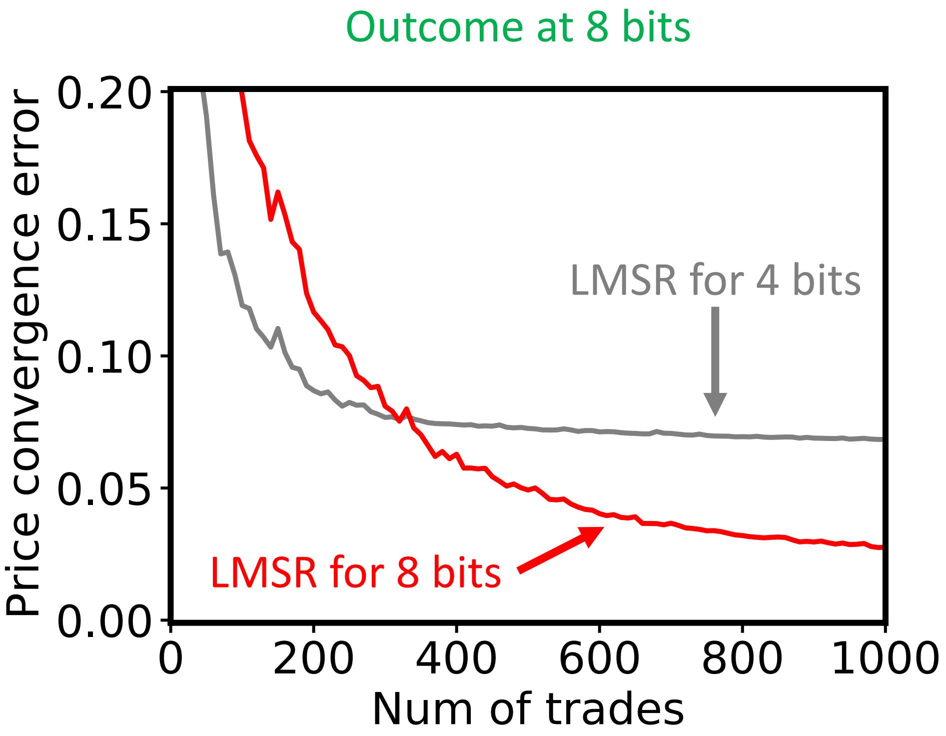
Contribution: Multi-resolution Linearly Constrained Market Maker

	Market Maker (MM)	Data Structure	Runtime of Market Operations	Worst-Case Loss for MM
previous work	Logarithmic market scoring rule (LMSR) [Hanson 2003]	array	$O(N)$ $\mathrm{N} = \#$ distinct outcomes	$\log(N)$
this work	Log-time LMSR MM	binary tree (adaptive)	$O(\log \mathrm{n}) \leq O(\log N)$ $n = \#$ distinct queries	$\log(N)$
	Multi-resolution linearly constrained MM (LCMM)	binary tree (static)	$O(\log N)$ $\mathrm{N} = \#$ distinct outcomes	constant

Log-time LMSR vs. Multi-resolution LCMM

- Simulate trading in prediction markets where the MM has a fixed budget.
- Evaluate how fast prices converge to reach “consensus”.





Log-time LMSR vs. Multi-resolution LCMM

- Compare to LCMM that equally splits the budget to two resolutions.
- **LCMM achieves the best of both worlds**:
  elicit forecasts at the finer level & obtain a fast convergence at the coarser level.

Recap & Summary

	Market Maker (MM)	Data Structure	Runtime of Market Operations	Worst-Case Loss for MM
previous work	Logarithmic market scoring rule (LMSR) [Hanson 2003]	array	$O(N)$ N = # distinct outcomes	$\log(N)$
this work	Log-time LMSR MM	binary tree (adaptive)	$O(\log \mathrm{n}) \leq O(\log N)$ n = # distinct queries	$\log(N)$
this work	Multi-resolution linearly constrained MM (LCMM)	binary tree (static)	$O(\log N)$ N = # distinct outcomes	constant

Thank you!