Proper Scoring Rules from Contracts to Markets

Xintong Wang[1], and Fang-Yi Yu[2]

[1]Rutgers University, [2]George Mason University

January 27, 2026

Outline

Current Progress

Elicit truthful reports

High quality information from the crowd

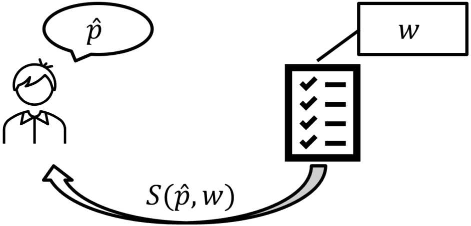
- Peer review at conferences
- Peer grading in classrooms
- Expert forecasting

- ○ **1.** Strong Reject 5%
- ○ **2.** Round 1 Reject 50%
- ○ **3.** Probable Eventual Reject 65%
- ○ **4.** Borderline (avoid using if possible) 70%
- ◉ **5.** Weak Accept 80 %
- ○ **6.** Accept 90%
- ○ **7.** Strong Accept 95%
- ○ **8.** Top (Best Paper Nomination) 99%
- ○ **9.** Very Top (Best Paper) 100%

Proper Scoring Rules: Binary

- Score an agent's forecast on a binary random variable on $\Omega = \{0, 1\}$
  - Agent reports a forecast $\hat{p} \in [0, 1]$
  - Principal and the agent observe the outcome $w \in \Omega$
  - Principal pays $S(\hat{p}, w)$ to the agent
- A scoring rule S is proper if for all $\hat{p}$

$$\mathbb{E}_{w \sim p}[S(p, w)] \geq \mathbb{E}_{w \sim p}[S(\hat{p}, w)]$$



Proper Scoring Rules

Definition

A scoring rule S is proper if for all $\hat{\boldsymbol{p}} \in \Delta_\Omega$,

$$S(\boldsymbol{p}, \boldsymbol{p}) \geq S(\hat{\boldsymbol{p}}, \boldsymbol{p})$$

where $S(\hat{\boldsymbol{p}}, \boldsymbol{p}) := \mathbb{E}_{w \sim \boldsymbol{p}}[S(\hat{\boldsymbol{p}}, w)]$, and strictly proper if the inequality is strict for all $\hat{\boldsymbol{p}} \neq \boldsymbol{p}$.

Score a forecast on a r.v. on Ω

- Report a forecast $\hat{\boldsymbol{p}} \in \Delta_\Omega$
- Observe the realization $w \in \Omega$
- Pay $S(\hat{\boldsymbol{p}}, w)$

Proper Scoring Rules

Definition

A scoring rule S is proper if for all $\hat{\boldsymbol{p}} \in \Delta_\Omega$,

$$S(\boldsymbol{p}, \boldsymbol{p}) \geq S(\hat{\boldsymbol{p}}, \boldsymbol{p})$$

where $S(\hat{\boldsymbol{p}}, \boldsymbol{p}) := \mathbb{E}_{w \sim \boldsymbol{p}}[S(\hat{\boldsymbol{p}}, w)]$, and strictly proper if the inequality is strict for all $\hat{\boldsymbol{p}} \neq \boldsymbol{p}$.

Examples of Proper Scoring Rules

- Log scoring rule: $S(\hat{\boldsymbol{p}}, w) = \ln \hat{p}(w)$
- Quadratic scoring rule: $S(\hat{\boldsymbol{p}}, w) = 2\hat{p}(w) - \|\hat{\boldsymbol{p}}\|^2 - 1$
- v-shaped for binary $\Omega = \{0, 1\}$: $S(\hat{p}, w) = (1 - c)1[p > c, w = 1] + c1[\hat{p} \leq c, w = 0]$

Current Progress

1. Proper Scoring Rules
   1.1 Definition of proper scoring rule
   1.2 Proper scoring rule = convex function
   - Application: Optimal scoring rules [Hartline et al., 2020]
   - Application: Optimal scoring rules Partial Knowledge setting
   1.3 Proper scoring rule = decision problem
2. Generalized Scoring Rules
3. Prediction Markets

What $S(\hat{\boldsymbol{p}}, w)$ are proper?

Theorem (Savage Representation)

The scoring rule S is (strictly) proper if and only if there exists a (strictly) convex function $G : \Delta_\Omega \to \mathbb{R}$ such that

$$S(\hat{\boldsymbol{p}}, w) = G(\hat{\boldsymbol{p}}) + \nabla G(\hat{\boldsymbol{p}}) \cdot (1_w - \hat{\boldsymbol{p}})$$

where ∇G is the (sub)gradient and 1_w is the distribution putting probability 1 on $w \in \Omega$.

Proper scoring rule = convex function

What $S(\hat{\boldsymbol{p}}, w)$ are proper?

Theorem (Savage Representation)

The scoring rule S is (strictly) proper if and only if there exists a (strictly) convex function $G : \Delta_\Omega \to \mathbb{R}$ such that

$$S(\hat{\boldsymbol{p}}, w) = G(\hat{\boldsymbol{p}}) + \nabla G(\hat{\boldsymbol{p}}) \cdot (1_w - \hat{\boldsymbol{p}})$$

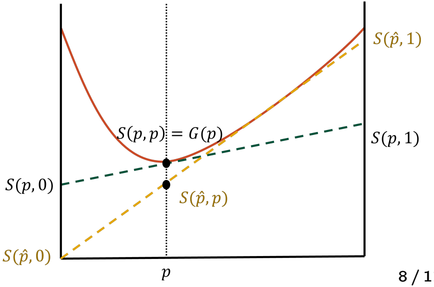
where ∇G is the (sub)gradient and 1_w is the distribution putting probability 1 on $w \in \Omega$.

[Proof of $\Leftarrow$]

$$\begin{aligned} S(\hat{\boldsymbol{p}}, \boldsymbol{p}) &= G(\hat{\boldsymbol{p}}) + \nabla G(\hat{\boldsymbol{p}}) \cdot (\mathbb{E}_{w \sim \boldsymbol{p}}[1_w] - \hat{\boldsymbol{p}}) \\ &= G(\hat{\boldsymbol{p}}) + \nabla G(\hat{\boldsymbol{p}}) \cdot (\boldsymbol{p} - \hat{\boldsymbol{p}}) . \end{aligned}$$

Because G is convex,

$$G(\hat{\boldsymbol{p}}) + \nabla G(\hat{\boldsymbol{p}}) \cdot (\boldsymbol{p} - \hat{\boldsymbol{p}}) \leq G(\boldsymbol{p}) = S(\boldsymbol{p}, \boldsymbol{p}).$$



What $S(\hat{\boldsymbol{p}}, w)$ are proper?

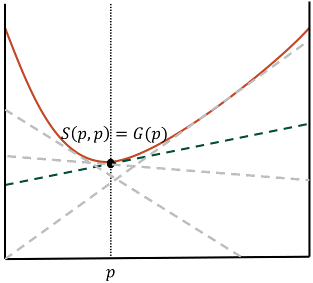
Theorem (Savage Representation)

The scoring rule S is (strictly) proper if and only if there exists a (strictly) convex function $G : \Delta_\Omega \to \mathbb{R}$ such that

$$S(\hat{\boldsymbol{p}}, w) = G(\hat{\boldsymbol{p}}) + \nabla G(\hat{\boldsymbol{p}}) \cdot (1_w - \hat{\boldsymbol{p}})$$

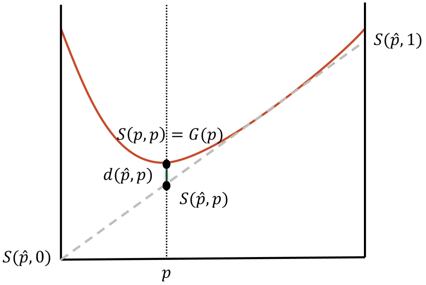
where ∇G is the (sub)gradient and 1_w is the distribution putting probability 1 on $w \in \Omega$.

[Proof of $\Rightarrow$] Let $G(\boldsymbol{p}) := S(\boldsymbol{p}, \boldsymbol{p})$. As $G(\boldsymbol{p}) = \max_{\hat{\boldsymbol{p}}} S(\hat{\boldsymbol{p}}, \boldsymbol{p})$, and $S(\hat{\boldsymbol{p}}, \boldsymbol{p})$ is affine in $\boldsymbol{p}$, $G(\boldsymbol{p})$ is convex. $S(\hat{\boldsymbol{p}}, \boldsymbol{p})$ is tangent to G at $\hat{\boldsymbol{p}}$, so $S(\hat{\boldsymbol{p}}, w) = G(\hat{\boldsymbol{p}}) + \nabla G(\hat{\boldsymbol{p}}) \cdot (1_w - \hat{\boldsymbol{p}})$ for some sub-gradient $\nabla G(\hat{\boldsymbol{p}})$.



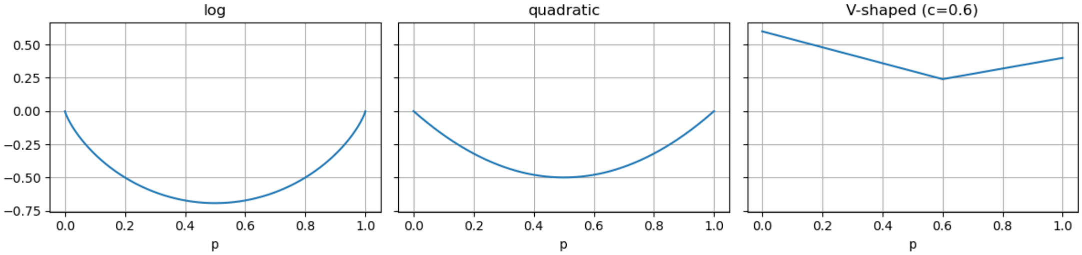
Given a proper scoring rule S,

- generalized entropy
  $G(\boldsymbol{p}) := S(\boldsymbol{p}, \boldsymbol{p}) = \sup_{\hat{\boldsymbol{p}}} S(\hat{\boldsymbol{p}}, \boldsymbol{p})$.
- divergence $d(\boldsymbol{q}, \boldsymbol{p}) := S(\boldsymbol{p}, \boldsymbol{p}) - S(\boldsymbol{q}, \boldsymbol{p})$
  - If S is strictly proper, $d(\boldsymbol{q}, \boldsymbol{p}) > 0$ unless $\boldsymbol{q} = \boldsymbol{p}$.
  - Generally not symmetric, $d(\boldsymbol{q}, \boldsymbol{p}) \neq d(\boldsymbol{p}, \boldsymbol{q})$.
  - also known as *Bregman divergence* with G,
    since $d(\boldsymbol{q}, \boldsymbol{p}) = G(\boldsymbol{p}) - G(\boldsymbol{q}) - \nabla G(\boldsymbol{q})(\boldsymbol{p} - \boldsymbol{q})$.



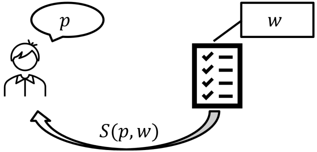
Examples of Information Measures and Divergence

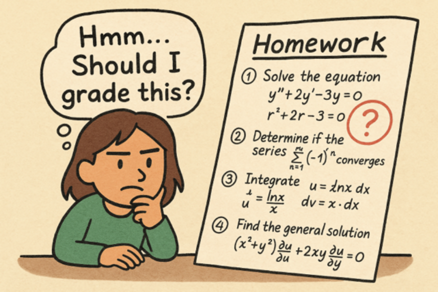
Scoring rules	$G(p)$	divergence $d(\hat{p}, p)$
Log	$p \ln p + (1-p)\ln(1-p)$	$\ln \frac{p}{\hat{p}} + (1-p)\ln \frac{1-\hat{p}}{1-p} = D_{KL}(\hat{p}, p)$
Quadratic	$-2p(1-p)$	$2(p-q)^2$
v-shaped	$c(1-p)1[p < c] + (1-c)p1[p \geq c]$	$\begin{cases} 0 & \text{if } p, q < c \text{ or } p, q \geq c \\ \lvert p - c \rvert & \text{otherwise} \end{cases}$



Peer review

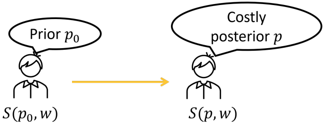
1. Principle announces S
2. Agent reports $\hat{p} \in [0, 1]$
3. Outcome $w \in \{0, 1\}$ reveals
4. Agent gets $S(\hat{p}, w)$

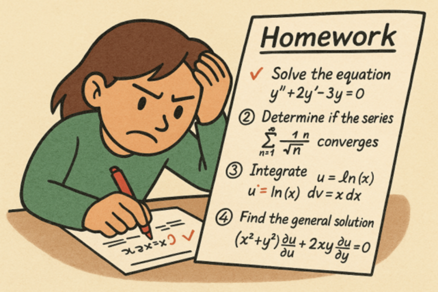




Peer review with effort

1. Principle announces S
2. Agent decides to acquire costly information P
3. Agent reports $\hat{p} \in [0, 1]$
4. Outcome $w \in \{0, 1\}$ reveals
5. Agent gets $S(\hat{p}, w)$

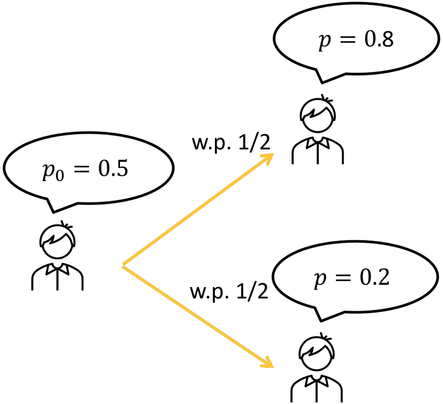




Incentivize costly forecasts

Peer review with effort

1. Principle announces S
2. Agent decides to acquire costly information P
3. Agent reports $\hat{p} \in [0, 1]$
4. Outcome $w \in \{0, 1\}$ reveals
5. Agent gets $S(\hat{p}, w)$



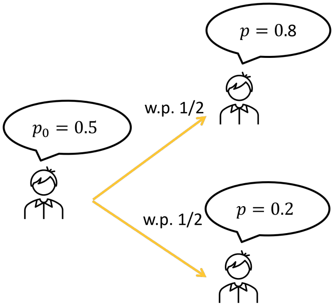
Incentivize costly forecasts

Given a joint distribution P between w and p, the expected payments before and after costly signal are

- Truthful prior: $G(p_0)$
- Truthful posterior: $\mathbb{E}_P[G(p)]$ where $\mathbb{E}_P[p] = p_0$
- Information gain: difference of payment

$$\mathbb{E}_P[G(p)] - G(p_0) = \mathbb{E}_P[G(p)] - G(\mathbb{E}_P[p])$$

The gap of Jensen's ineq. = convexity at p_0



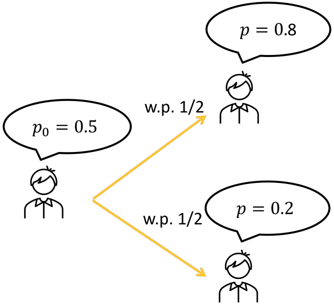
Optimization of scoring rule: Model [Hartline et al., 2020]

Model

Given an information structure P on (w, p), design "bounded" scoring rule S with G so that maximize the expected gain

$$\max_G \mathbb{E}_P[G(p)] - G(\mathbb{E}_P[p]) \text{ such that } G \text{ is convex and bounded}$$

1. Bounded ex-post payment [Hartline et al., 2020]: $0 \leq S(p, w) \leq 1$.
2. Bounded expected payment [Chen and Yu, 2021]: $0 \leq G \leq 1$.

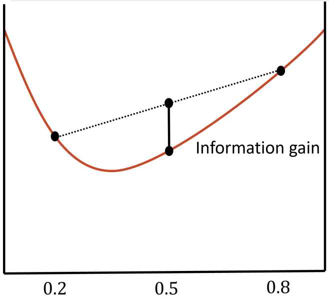


Optimization of scoring rule: Theorem

Theorem

v-shaped scoring rules are optimal. Let p_0 be the prior of the $w \in \{0,1\}$, and $x \mapsto \max\{a(x-x_0)+c, b(x-x_0)+c\}$ be a v-shaped function with (x_0, a, b, c).

- *A v-shaped function with $(p_0, \frac{-1}{p_0}, \frac{1}{1-p_0}, 0)$ is optimal for the ex-ante setting.*
- *A v-shaped function with $(p_0, \frac{-1}{2\max\{p_0, 1-p_0\}}, \frac{1}{2\max\{p_0, 1-p_0\}}, \frac{1}{2})$ is optimal for the ex-post setting.*

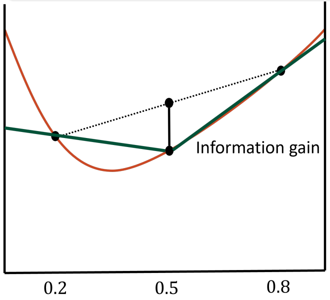


Optimization of scoring rule: Theorem

Theorem

v-shaped scoring rules are optimal. Let p_0 *be the prior of the* $w \in \{0,1\}$*, and* $x \mapsto \max\{a(x-x_0)+c, b(x-x_0)+c\}$ *be a v-shaped function with* (x_0, a, b, c).

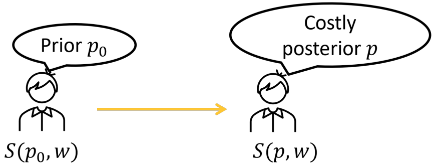
- *A v-shaped function with* $(p_0, \frac{-1}{p_0}, \frac{1}{1-p_0}, 0)$ *is optimal for the ex-ante setting.*
- *A v-shaped function with* $(p_0, \frac{-1}{2\max\{p_0, 1-p_0\}}, \frac{1}{2\max\{p_0, 1-p_0\}}, \frac{1}{2})$ *is optimal for the ex-post setting.*



How can we handle unknown P

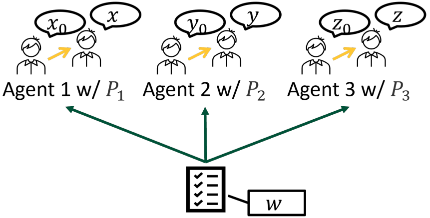
Peer review with effort

1. Principle announce S
2. Agent decides to acquire costly information P
3. Agent reports $\hat{p} \in [0, 1]$
4. Outcome reveals $w \in \{0, 1\}$
5. Agent gets $S(\hat{p}, w)$

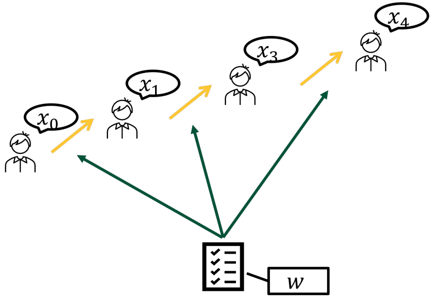


Multiple possible information structures $\mathcal{P} = \{P_1, \ldots\}$

Heterogeneous agents

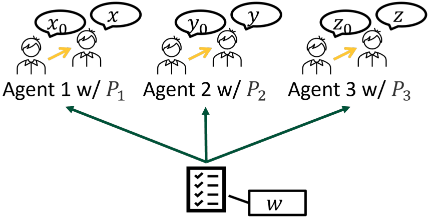


Sequential learning

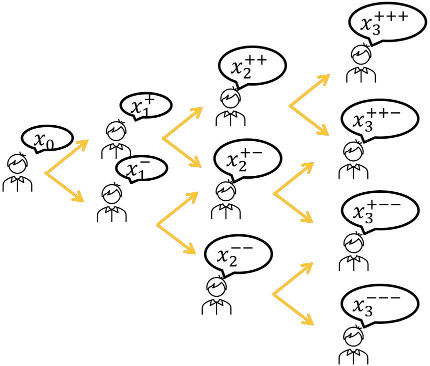


Multiple possible information structures $\mathcal{P} = \{P_1, \ldots\}$

Heterogeneous agents



Sequential learning

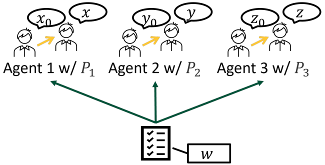


Optimization of scoring rule: Model [Chen and Yu, 2021]

Model

Given a collection of information structure $\mathcal{P}$ on (w, p), design "bounded" scoring rule S with G so that maximizes the expected gain

$$\max_{G} \min_{P \in \mathcal{P}} \mathbb{E}_P[G(p)] - G(\mathbb{E}_P[p]) \text{ such that } G \text{ is convex and bounded}$$

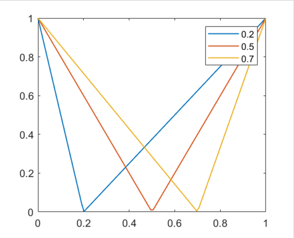


Optimization of scoring rule: Results

Different $\mathcal{P}$ leads to different optimal scoring rules

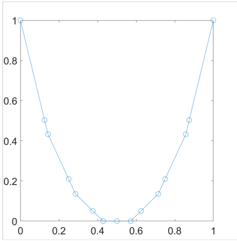
1. Singleton: a v-shaped G is optimal $\rightarrow$ turning point at prior
2. Finite $\mathcal{P}$: an efficient algorithm and is piecewise linear is optimal $\rightarrow$ turning points at support of all information structures.

Singleton $\mathcal{P}$



V-shape

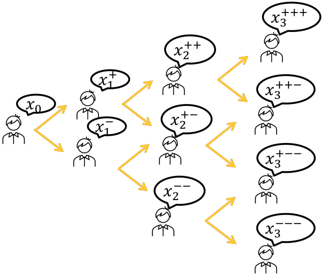
Finite $\mathcal{P}$

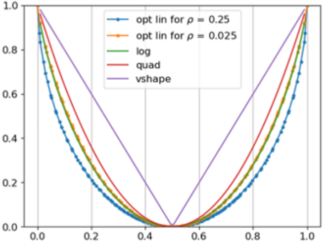


Piecewise linear

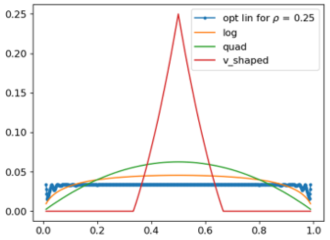
Optimization of scoring rule: Simulations

Log scoring rule perform well under Beta-Bernoulli setting





(a) Associated convex functions



(b) Information gain with $\rho = 0.25$.

Current Progress

Bayesian Decision Problem

A *decision problem* $(\mathcal{A}, \Omega, u)$ consists of an action space (decisions) $\mathcal{A}$, an outcome space Ω, and a value function $u : \mathcal{A} \times \Omega \to \mathbb{R}$. An agent chooses an action based on belief $\boldsymbol{p} \in \Delta_\Omega$ of the outcome w to maximize the expected utility.

- Given an action a, the agent gets $u(a, w)$ under an outcome w and $u(a, \boldsymbol{p}) := \mathbb{E}_{w \sim \boldsymbol{p}}[u(a, w)]$ in expectation.
- $a_{\boldsymbol{p}} \in \mathcal{A}$ is a Bayes act/best response to $\boldsymbol{p}$ if for all a, $u(a_{\boldsymbol{p}}, \boldsymbol{p}) \geq u(a, \boldsymbol{p})$, and

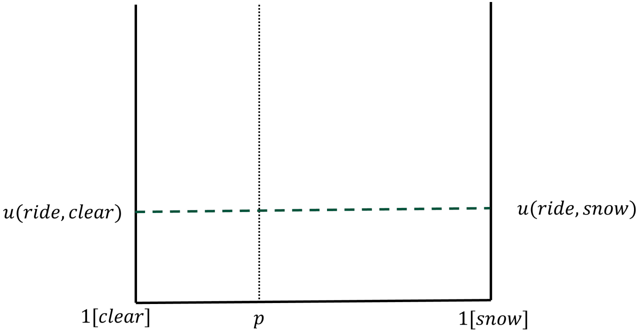
$$U(\boldsymbol{p}) := \max_{a \in \mathcal{A}} u(a, \boldsymbol{p})$$

Example of Decision Problem[1]

A journey through Rutgers

- $\Omega = \{\text{clear}, \text{snow}\}$,
- $\mathcal{A} = \{\text{ride}, \text{walk}\}$.
- value function
  - If we ride

$$u(\text{ride}, \text{clear}) = 4,$$
$$u(\text{ride}, \text{snow}) = 4.$$



$$u(\text{ride}, p) = 4$$

[1] Credit: Adapted from Bo Waggoner's slides.

Example of Decision Problem[1]

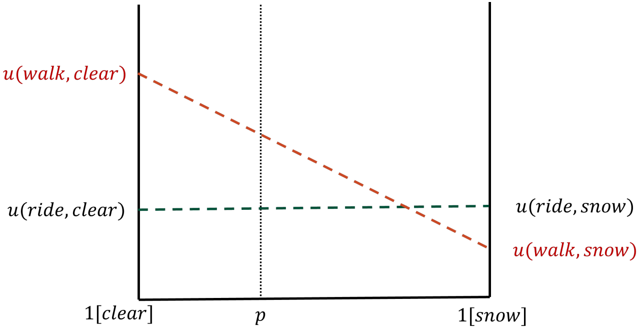
A journey through Rutgers

- $\Omega = \{\text{clear}, \text{snow}\}$,
- $\mathcal{A} = \{\text{ride}, \text{walk}\}$.
- value function
  - If we ride

$$u(\text{ride}, \text{clear}) = 4,$$
$$u(\text{ride}, \text{snow}) = 4.$$

  - If we walk

$$u(\text{walk}, \text{clear}) = 10,$$
$$u(\text{ride}, \text{snow}) = 2.$$



$$u(\text{walk}, p) = 10 - 8p$$

[1] Credit: Adapted from Bo Waggoner's slides.

Example of Decision Problem[1]

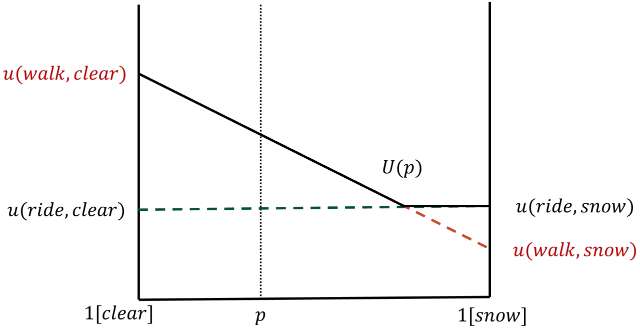
A journey through Rutgers

- $\Omega = \{\text{clear}, \text{snow}\}$,
- $\mathcal{A} = \{\text{ride}, \text{walk}\}$.
- value function
  - If we ride

$$u(\text{ride}, \text{clear}) = 4,$$
$$u(\text{ride}, \text{snow}) = 4.$$

  - If we walk

$$u(\text{walk}, \text{clear}) = 10,$$
$$u(\text{ride}, \text{snow}) = 2.$$



$$U(p) = \max_{a=\text{ride, walk}} u(a, p) = \max\{4, 10 - 8p\}$$

[1] Credit: Adapted from Bo Waggoner's slides.

Example of Decision Problem[1]

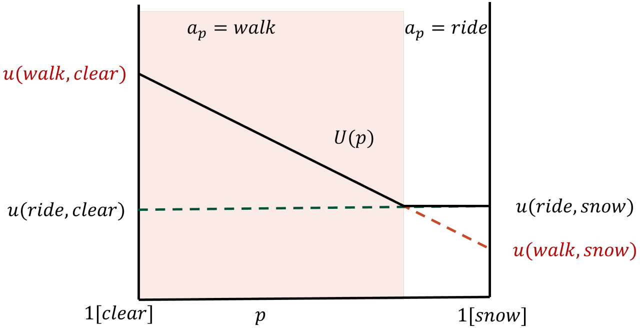
A journey through Rutgers

- $\Omega = \{\text{clear}, \text{snow}\}$,
- $\mathcal{A} = \{\text{ride}, \text{walk}\}$.
- value function
  - If we ride

$$u(\text{ride}, \text{clear}) = 4,$$
$$u(\text{ride}, \text{snow}) = 4.$$

  - If we walk

$$u(\text{walk}, \text{clear}) = 10,$$
$$u(\text{ride}, \text{snow}) = 2.$$



$$a_p = \begin{cases} \text{walk} & \text{if } p < 3/4 \\ \text{ride} & \text{otherwise.} \end{cases}$$

[1] Credit: Adapted from Bo Waggoner's slides.

Proper Scoring Rules = Decision Problem

Theorem

For any decision problem $(\mathcal{A}, \Omega, u)$ *there exists a proper scoring rule* $S : \Delta_\Omega \times \Omega \to \mathbb{R}$ *with* G *so that for all belief* $\boldsymbol{p}$

$$G(\boldsymbol{p}) = U(\boldsymbol{p})$$

Proof. Set $S(\hat{\boldsymbol{p}}, w) := u(a_{\hat{\boldsymbol{p}}}, w)$ and use revelation principal. □

Proper Scoring Rules = Decision Problem

Theorem

For any decision problem $(\mathcal{A}, \Omega, u)$ there exists a proper scoring rule $S : \Delta_\Omega \times \Omega \to \mathbb{R}$ with G so that for all belief $\boldsymbol{p}$

$$G(\boldsymbol{p}) = U(\boldsymbol{p})$$

Proof. Set $S(\hat{\boldsymbol{p}}, w) := u(a_{\hat{\boldsymbol{p}}}, w)$ and use revelation principal. □

Note that a scoring rule is a special case of decision problem where the action space $\mathcal{A} = \Delta_\Omega$.

Proper Scoring Rules = Decision Problem

Theorem

For any decision problem $(\mathcal{A}, \Omega, u)$ *there exists a proper scoring rule* $S : \Delta_\Omega \times \Omega \to \mathbb{R}$ *with* G *so that for all belief* $\boldsymbol{p}$

$$G(\boldsymbol{p}) = U(\boldsymbol{p})$$

Decision problem

- $\Omega = \{\text{clear}, \text{snow}\}$,
- $\mathcal{A} = \{\text{ride}, \text{walk}\}$, and
- $$\begin{aligned} u(\text{ride}, \text{clear}) &= 4, \\ u(\text{ride}, \text{snow}) &= 4, \\ u(\text{walk}, \text{clear}) &= 10, \\ u(\text{ride}, \text{snow}) &= 2. \end{aligned}$$

Proper Scoring rule

- $\Omega = \{\text{clear}, \text{snow}\}$,
- $\Delta_\Omega = [0, 1]$ probability of snow
- $S(\hat{p}, w) = u(a_{\hat{p}}, w) =$
$$\begin{cases} 10 & \text{if } \hat{p} \leq 3/4, w = \text{clear} \\ 2 & \text{if } \hat{p} \leq 3/4, w = \text{snow} \\ 4 & \text{otherwise.} \end{cases}$$

v-shaped scoring rule!

Applications: Monotonicity of information

Given a decision problem $(\mathcal{A}, \Omega, u)$, which do you prefer?

- a signal s (e.g., COVID test), and get a posterior of the outcome $\boldsymbol{p}^s = \Pr[\mathrm{w} \mid s]$, or
- prior $\boldsymbol{p} = \Pr[\mathrm{w}]$.

Theorem (Information never harms)

$$\mathbb{E}[u(a_{\boldsymbol{p}^s}, w)] \geq \mathbb{E}[u(a_{\boldsymbol{p}}, w)]$$

Proof.

$$\begin{aligned}
\mathbb{E}[u(a_{\boldsymbol{p}^s}, w)] &= \mathbb{E}[G(\boldsymbol{p}^s)] && \text{(decision problem = proper scoring rule)} \\
&\geq G(\mathbb{E}\boldsymbol{p}^s) && (G \text{ is convex}) \\
&= \mathbb{E}[u(a_{\boldsymbol{p}}, w)]. && \text{(decision problem = proper scoring rule)}
\end{aligned}$$

□

Applications: U-calibration [Kleinberg et al., 2023]

How can we measure the quality of a sequence of forecasts and outcomes (p_t, w_t) for agents with unknown decision problems?

- Given a decision problem u, the regret of following (best responding) the forecasts is

$$Reg_u = \max_a \sum_t u(a, w_t) - \sum_t u(a_{p_t}, w_t) = \max_q \sum_t S(q, w_t) - \sum_t S(p_t, w_t)$$

- U-calibration error is the worst regret on all bounded decision problems $\mathcal{U}$,

$$UCal = \sup_{u \in \mathcal{U}} Reg_u = \sup_{S \text{ bounded proper}} \left[\max_q \sum_t S(q, w_t) - \sum_t S(p_t, w_t) \right].$$

- U-calibration$\neq \ell_1$-Calibration
  - ℓ_1-calibration punishes everywhere
  - U-calibration is budgeted (recall that for the scoring rule design: G cannot be too curved). In particular, U-calibration $\approx$ V-calibration.

Scoring rule design = mechanism design[2]

Scoring rule design

$$\max_{\text{scoring rule}} \mathbb{E}[\text{ objective}]$$

s.t scoring rule is proper and bounded

A scoring rule is proper iff

1. utility of agent's forecast is convex
2. score evaluates state on supporting plane of utility

Mechanism Design

$$\max_{\text{mechanism}} \mathbb{E}[\text{ objective}]$$

s.t mechanism is i.c. and feasible

A mechanism incentive compatible iff [Rochet, 1985]

1. utility of agent's forecast is convex
2. allocation is sub-gradient of utility (with payment, gives supporting hyperplane)

[2] Credit: Adapted from Jason Hartline's slides. Also check out [Frongillo and Kash, 2014]

Current Progress

Beyond scoring forecast

	Scoring rule S	Decision problem u	General loss function ℓ
Report	forecast $\hat{\boldsymbol{p}} \in \Delta_\Omega$	action $a \in \mathcal{A}$	$r \in \mathcal{R}$
Observe	outcome $w \in \Omega$	outcome $w \in \Omega$	observation $y \in \mathcal{Y}$
Reward	$S(\hat{\boldsymbol{p}}, w)$	$u(a, w)$	$-\ell(r, y)$

- Property elicitation: Can we directly elicit a specific property of a distribution (e.g., quantile, mean, variance)?
- Surrogate scoring rule, peer prediction: Rather than the true outcome w, can we use a noisy or stochastically related observation?

Property Elicitation: Definition

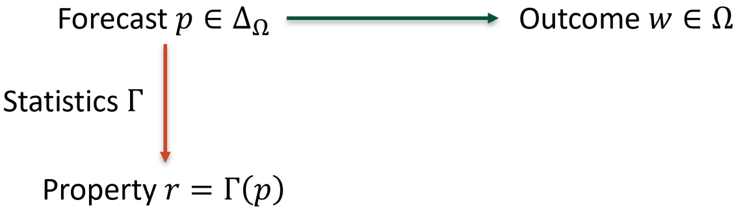
Definition

A *property/statistic* is a function $\Gamma : \Delta_\Omega \to \mathcal{R}$. A (generalized) scoring rule $S : \mathcal{R} \times \Omega \to \mathbb{R}$ *elicits* Γ if

$$\Gamma(\boldsymbol{p}) = \arg\max_{r \in \mathcal{R}} \mathbb{E}_{w \sim \boldsymbol{p}} S(r, w).$$

Moreover, Γ is *elicitable* if there exists S that elicit it.

Goal: Ask for statistics rather than full distributions, e.g., mean, variance, median, and ensure that $S(\Gamma(\boldsymbol{p}), \boldsymbol{p}) \geq S(r, \boldsymbol{p})$ for all $r \in \mathcal{R}$.



Property Elicitation: Threshold property

Threshold property

- (Forecast) What is the probability of snow tomorrow?
- (Property) Is the probability of snow larger than 3/4? $\Gamma(p) = 1[p > 3/4]$

Property Elicitation: Threshold property

Threshold property

- (Forecast) What is the probability of snow tomorrow?
- (Property) Is the probability of snow larger than 3/4? $\Gamma(p) = 1[p > 3/4]$

create a decision problem to score property (do you ride?)

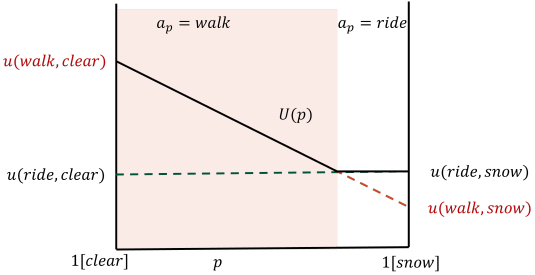
Threshold property

- (Forecast) What is the probability of snow tomorrow?
- (Property) Is the probability of snow larger than 3/4? $\Gamma(p) = 1[p > 3/4]$

create a decision problem to score property (do you ride?)

Decision problem u

- $\Omega = \{\text{clear}, \text{snow}\}$,
- $\mathcal{A} = \{\text{ride}, \text{walk}\}$, and
- $\begin{cases} u(\text{ride}, \text{clear}) = & 4, \\ u(\text{ride}, \text{snow}) = & 4, \\ u(\text{walk}, \text{clear}) = & 10, \\ u(\text{ride}, \text{snow}) = & 2. \end{cases}$



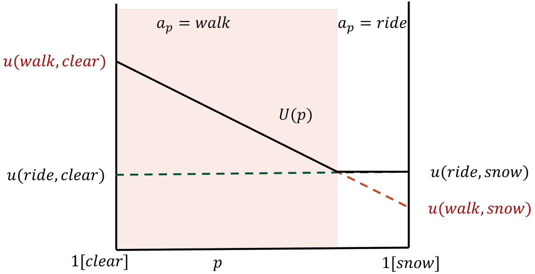
Threshold property

- (Forecast) What is the probability of snow tomorrow?
- (Property) Is the probability of snow larger than 3/4? $\Gamma(p) = 1[p > 3/4]$

create a decision problem to score property (do you ride?)

Scoring rule for property S

- $\Omega = \{\text{clear}, \text{snow}\}$,
- $\mathcal{A} = \{0, 1\}$, and
- $$\begin{cases} S(1, \text{clear}) = & 4, \\ S(1, \text{snow}) = & 4, \\ S(0, \text{clear}) = & 10, \\ S(1, \text{snow}) = & 2. \end{cases}$$



Elicit General threshold property

Threshold property

Is the probability of snow larger than c? ($\Gamma(p) = 1[p > c]$ and $\mathcal{R} = \{0, 1\}$.)

$$\begin{cases} S(1, \text{clear}) = & 0, \\ S(1, \text{snow}) = & 1 - c, \\ S(0, \text{clear}) = & c, \\ S(1, \text{snow}) = & 0. \end{cases}$$

v-shaped binary Ω: $S(\hat{p}, w) = (1 - c)1[p > c, w = 1] + c1[\hat{p} \leq c, w = 0]$

Elicit Mode

Mode

$\Gamma(p) = \arg\max_w p(w)$ and $\mathcal{R} = \Omega$.

- Idea: create a decision problem to score property
- $S(r, w) = 1[r = w]$

Elicit Mean

Mean of real-valued random variable

How much snow do you expect will fall tomorrow? ($\Gamma(\boldsymbol{p}) = \mathbb{E}_{w\sim\boldsymbol{p}}[w]$ and $\mathcal{R} = \mathbb{R}$.)

- Idea: create a loss function to score property
- As the expectation minimizes the squared loss, we can take $S(r, w) = -\|r - w\|^2$ that elicits mean.

Property for real-valued random variable

- We can derive scoring rules from loss functions in ML

Statistic/Property Γ	scoring rule for Γ	loss function
Mean	$-(r-w)^2$	square loss
Median	$-\lvert r-w\rvert$	absolute
α-quantile	$-(r-w)(1[r \geq w]-\alpha)$	Pinball
Mode	$1[r=w]$	zero-one loss

- Are all property elicitable? The variance is not (directly) elicitable in general.

Proof.[3] Consider a Bernoulli on $\{0,1\}$ with p. Suppose that a scoring rule S elicits the variance.

- For $p=1$, $w=1$ surely and the optimal report is 0, $S(r,1) \leq S(0,1)$ for all r.
- For $p=0$, $w=0$ surely, and $S(r,1) \leq S(0,0)$ for all r.

$S(r,p) = p \cdot S(r,1) + (1-p) \cdot S(r,0) \leq S(0,p)$ for all r and p. □

[3] Adapted from Bo's blog

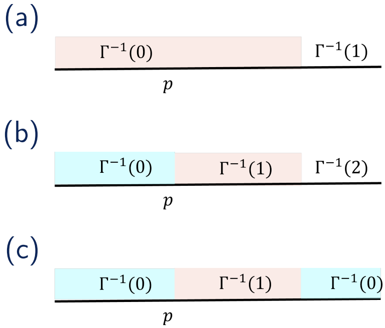
Elicit Finite-valued properties [Lambert and Shoham, 2009]

- Given a property $\Gamma : \Delta_\Omega \to \mathcal{R}$, a level set consists of distributions that have the same correct answer $\Gamma^{-1}(r)$.

$\Gamma(p) = 1[p > 3/4]$

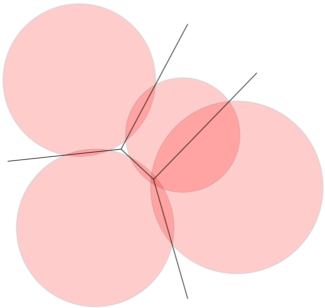
Elicit Finite-valued properties [Lambert and Shoham, 2009]

- Given a property $\Gamma : \Delta_\Omega \to \mathcal{R}$, a level set consists of distributions that have the same correct answer $\Gamma^{-1}(r)$.
- Which do you think are elicitable?



Elicit Finite-valued properties [Lambert and Shoham, 2009]

- Given a property $\Gamma : \Delta_\Omega \to \mathcal{R}$, a level set consists of distributions that have the same correct answer $\Gamma^{-1}(r)$.
- Which do you think are elicitable?
- Γ is elicitable if and only if Γ is power diagram (weighted vornoi diagram): Given a set of points $\boldsymbol{c}_i \in \Delta_\Omega$ and weights $d_i \in \mathbb{R}$, $cell_i := \{\boldsymbol{p} : \|\boldsymbol{c}_i - \boldsymbol{p}\|^2 - d_i \leq \|\boldsymbol{c}_j - \boldsymbol{p}\|^2 - d_j, \forall j\}$.)



Elicit Finite-valued properties [Lambert and Shoham, 2009]

Theorem

A finite-valued property Γ is elicitable if and only if $\{\Gamma^{-1}(r) : r \in \mathcal{R}\}$ is a power diagram of Δ_Ω for some set of weighted sites $(\boldsymbol{c}_r, d_r)_{r\in\mathcal{R}}$.

Proof. $\Leftarrow$) Given a power diagram with $(\boldsymbol{c}_s, d_s)_{s\in\mathcal{R}}$, let $S(r, w) := 2\langle 1_w, \boldsymbol{c}_r\rangle + d_r - \|\boldsymbol{c}_r\|^2$ for all $r \in \mathcal{R}$ and $w \in \Omega$. We show the score elicit the following property

$$\Gamma(\boldsymbol{p}) = \{r : \|\boldsymbol{c}_r - \boldsymbol{p}\|^2 - d_r \le \|\boldsymbol{c}_s - \boldsymbol{p}\|^2 - d_s, \forall s\}.$$

For all $r, s \in \mathcal{R}$ and $\boldsymbol{p}$ with $r \in \Gamma(\boldsymbol{p})$ and $s \notin \Gamma(\boldsymbol{p})$

$$\begin{aligned}\mathbb{E}_{w\sim\boldsymbol{p}}[S(s, w)] =& 2\langle \boldsymbol{c}_s, \boldsymbol{p}\rangle + d_s - \|\boldsymbol{c}_s\|^2 \\ =& \|\boldsymbol{p}\|^2 - \|\boldsymbol{p} - \boldsymbol{c}_s\|^2 + d_s \\ <& \|\boldsymbol{p}\|^2 - \|\boldsymbol{p} - \boldsymbol{c}_r\|^2 + d_r = \mathbb{E}_{w\sim\boldsymbol{p}}[S(r, w)].\end{aligned}$$

□

Elicit Finite-valued properties [Lambert and Shoham, 2009]

Theorem

A finite-valued property Γ is elicitable if and only if $\{\Gamma^{-1}(r) : r \in \mathcal{R}\}$ is a power diagram of Δ_Ω for some set of weighted sites $(\boldsymbol{c}_r, d_r)_{r\in\mathcal{R}}$.

Proof. $\Rightarrow$) If S elicits Γ, let $\boldsymbol{c}_r := \frac{1}{2}(S(r,w))_{w\in\Omega} \in \mathbb{R}^{|\Omega|}$, and $d_r = \|\boldsymbol{c}_r\|^2 \in \mathbb{R}$. Now we show $r \in \Gamma(\boldsymbol{p})$ if and only if $\|\boldsymbol{c}_r - \boldsymbol{p}\|^2 - d_r \leq \|\boldsymbol{c}_s - \boldsymbol{p}\|^2 - d_s, \forall s$. For all r, s and $\boldsymbol{p}$ with $r \in \Gamma(\boldsymbol{p})$,

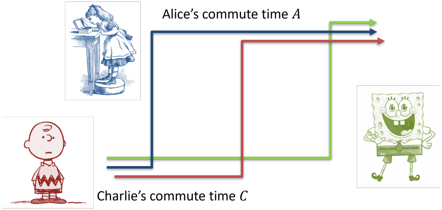
$$\begin{aligned}
\|\boldsymbol{c}_s - \boldsymbol{p}\|^2 - d_s &= \|\boldsymbol{c}_s\|^2 - 2\langle \boldsymbol{c}_s, \boldsymbol{p}\rangle + \|\boldsymbol{p}\|^2 - d_s \\
&= -2\langle \boldsymbol{c}_s, \boldsymbol{p}\rangle + \|\boldsymbol{p}\|^2 && (d_r = \|\boldsymbol{c}_r\|^2) \\
&= -\mathbb{E}_{w\sim\boldsymbol{p}}[S(s,w)] + \|\boldsymbol{p}\|^2 && (\mathbb{E}_{w\sim\boldsymbol{p}}[S(s,w)] = 2\langle \boldsymbol{c}_s, \boldsymbol{p}\rangle) \\
&\geq -\mathbb{E}_{w\sim\boldsymbol{p}}[S(r,w)] + \|\boldsymbol{p}\|^2 && (\mathbb{E}_{w\sim\boldsymbol{p}}[S(r,w)] \geq \mathbb{E}_{w\sim\boldsymbol{p}}[S(s,w)]) \\
&= \|\boldsymbol{c}_r - \boldsymbol{p}\|^2 - d_r
\end{aligned}$$

□

Current Progress

Application: Peer prediction

- Proper scoring rules require the outcome w which is always not observable
  - Subjective: Are you happy? Do prefer ChatGPT or Gemini?
  - Private: What is your commute time?
- Peer prediction: As agents' signals are often dependent, we can use their report to elicit agents' truthful reports.



Peer prediction through proper scoring rule [Miller et al., 2005]

Alice and Bob have signals in $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ respectively jointly sampled from P.

- Alice and Bob report $\hat{x}$ and $\hat{y}$.
- Compute their posteriors $P(\cdot \mid \hat{x}), P(\cdot \mid \hat{y})$ on a scoring rule S,

$$M_A(\hat{x}, \hat{y}) = S(P(\cdot \mid \hat{x}), \hat{y}) \text{ and } M_B(\hat{x}, \hat{y}) = S(P(\cdot \mid \hat{y}), \hat{x}).$$

- Pros and cons
  - Truthful: Ensure truth-telling is a Bayesian Nash equilibrium[4], since S is proper

$$\mathbb{E}[M_A(x, y)] = S(\boldsymbol{p}, \boldsymbol{p}) \geq S(\hat{\boldsymbol{p}}, \boldsymbol{p}) = \mathbb{E}[M_A(\hat{x}, y)] \text{ with } \boldsymbol{p} = P(\cdot \mid x), \hat{\boldsymbol{p}} = P(\cdot \mid \hat{x})$$

  - Minimal: Agents only report their signals.
  - Not detailed-free: Require the knowledge of P.

[4] The truth-telling is a strict BNE when P is stochastic relevant.

Three tricks

Can we relax the knowledge of P?

1. Partial knowledge: restrict the possible P to a subset $\mathcal{P}$, e.g., self dominating, self-predicting
2. Non-minimal: Ask agents to report not only signal but forecast or second order forecast [Prelec, 2004]
3. Learn P from iid reports+DPI [Kong and Schoenebeck, 2019, Schoenebeck and Yu, 2020] or LLM [Lu et al., 2024]

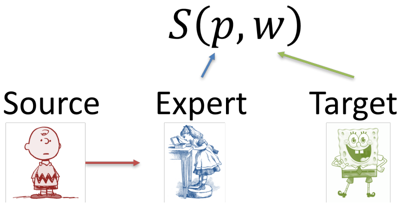
Peer Prediction and property elicitation [Frongillo and Witkowski, 2017]

Can we characterize all truthful minimal mechanisms M under $\mathcal{P}$?

- Truthful reporting is a property of posterior $\Gamma(\boldsymbol{p}) = x$ if and only if $\boldsymbol{p} \in D_x := \{P(\cdot \mid x) : P \in \mathcal{P}\} \subseteq \Delta_{\mathcal{Y}}$.
- Example: Output agreement algorithm
  - $\mathcal{X} = \mathcal{Y} = \{0, 1\}$
  - $\mathcal{P}$ consists of self-dominance distribution $P(z \mid z) > P(z' \mid z)$ for all $z, z' \in \{0, 1\}$.
  - $D_1 = \{p : p > (1 - p)\}$ and $D_0 = \{p : p < 1/2\}$
  - $\Gamma(p) = 1[p > 1/2]$ $=$ mode

Non-minimal peer prediction mechanism [Schoenebeck and Yu, 2023]

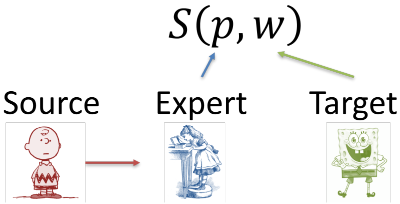
- Given a proper scoring rule, agents can play one of three roles
  - Expert: makes prediction.
  - Source: provides information to the expert.
  - Target: reports his signal and get predicted.
- We can design mechanisms by randomize agent's rules.



Source Differential peer prediction mechanism

Given a proper scoring rule S, in a source-DPP, three agents play one of three roles

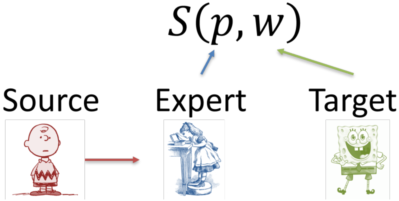
- Expert makes predictions p, p^+ and gets sum of the scores $S(p,t) + S(p^+,t)$
- Source provides signal to improve p^+ and get the difference $S(p^+,t) - S(p,t)$
- Target reports signal t and gets zero



Target Differential peer prediction mechanism

Given a *log scoring rule*, in target-DPP, three agents play one of three roles

- Expert makes two predictions and gets sum of the scores
- Source provides information for the second prediction and gets zero
- Target gets the difference



Connection to other peer prediction mechanism

Theorem ([Schoenebeck and Yu, 2023])

Source and Target-DPP are strongly truthful:

- *Truth-telling is a strict Bayesian Nash equilibrium.*
- *Truth-telling has the highest total payment (strictly better than non-permutation ones')*

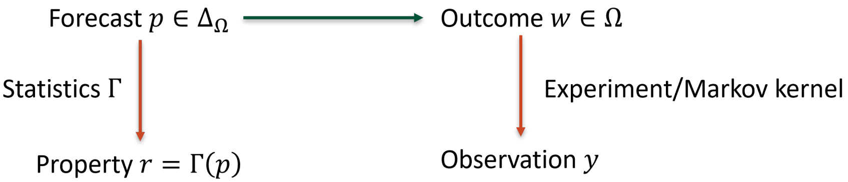
New view point of BTS [Prelec, 2004]:

- Everyone plays the target and also provides the first prediction.
- We can learn an improved prediction if there are many symmetric agents.

Current Progress

Beyond scoring forecast

- Proper scoring rule: score a forecast $\hat{\boldsymbol{p}} \in \Delta_\Omega$ using the outcome $w \in \Omega$
- Property elicitation: score a property $r \in \mathcal{R}$ using the outcome
- Do we need direct access to the true outcome w?



Surrogate scoring rule [Liu et al., 2023]

Can we incentivize high-quality prediction when the ground truth is unavailable?

- Motivation: "How likely a study can be replicated?"
  - Forecasters are asked to provide a probabilistic prediction.
  - The SCORE program crowdsourced this question for 3000 studies to hundreds of researchers, while only a small fraction will have a real replication test.
  - We may use other's report to derive a noisy ground truth.

Article | Open access | Published: 19 November 2024

Examining the replicability of online experiments selected by a decision market

Surrogate scoring rule

- Idea: We can treat an observation $y \in \mathcal{Y}$ as surrogate of w if we know the conditional probability of y given w $\boldsymbol{T} \in \mathbb{R}^{\Omega \times |\mathcal{Y}|}$.
- Surrogate scoring rule: For all proper scoring rule $S : \Delta_\Omega \times \Omega \to \mathbb{R}$ and invertible $\boldsymbol{T}$,

$$\tilde{S}(\hat{\boldsymbol{p}}, y) = \sum_z \boldsymbol{T}^{-1}(y, z) S(\hat{\boldsymbol{p}}, z)$$

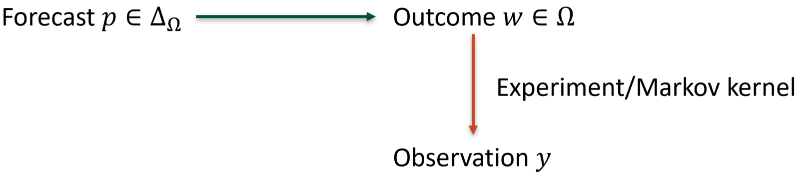
- If $\mathcal{Y} = \Omega = \{0, 1\}$ with $\Pr[y = 1|w = 0] = e^-$ and $\Pr[y = 0|w = 1] = e^+$,

$$\begin{aligned}
\tilde{S}(\hat{p}, 0) &= \frac{1}{1 - e^- - e^+} \left((1 - e^-)S(\hat{p}, 0) - e^+ S(\hat{p}, 1)\right) \\
\tilde{S}(\hat{p}, 1) &= \frac{1}{1 - e^- - e^+} \left(-e^+ S(\hat{p}, 0) + (1 - e^+)S(\hat{p}, 1)\right)
\end{aligned}$$

Surrogate scoring rule

Theorem

If $\boldsymbol{T} \in \mathbb{R}^{\Omega \times |\mathcal{Y}|}$ *has full row rank, the expectation of* $\tilde{S}(\hat{\boldsymbol{p}}, \cdot) = \boldsymbol{T}^{-1} S(\hat{\boldsymbol{p}}, \cdot)$ *equals* $\mathbb{E}_{w \sim \boldsymbol{p}}[S(\hat{\boldsymbol{p}}, w)]$ *for all* $\boldsymbol{p}$ *and* $\hat{\boldsymbol{p}}$.

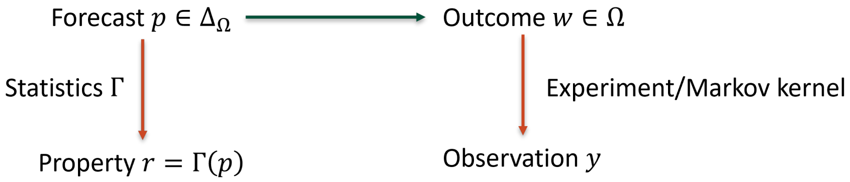


Proof. Because $\boldsymbol{T}^\top \Pr[w = \cdot] = Pr[y = \cdot]$ and proper scoring rules S are affine in the outcome space,

$$\mathbb{E}[\tilde{S}(\hat{\boldsymbol{p}}, y)] = \langle Pr[y = \cdot], \tilde{S}(\hat{\boldsymbol{p}}, \cdot)\rangle = \langle \boldsymbol{T}^\top Pr[w = \cdot], \boldsymbol{T}^{-1} S(\hat{\boldsymbol{p}}, \cdot)\rangle = \mathbb{E}[S, \hat{\boldsymbol{p}}, w)] \qquad \square$$

Surrogate scoring rule and property elicitation

- Backward correction: change the observation y to mimic w.[5]
- Forward correction: treat the forecast $\boldsymbol{p}$ of w as a property of observation y where $\Gamma(\boldsymbol{q}_y) = \boldsymbol{T}\boldsymbol{q}_y = \boldsymbol{p}$, and pay $S(\Gamma^{-1}(\hat{\boldsymbol{p}}, y)$

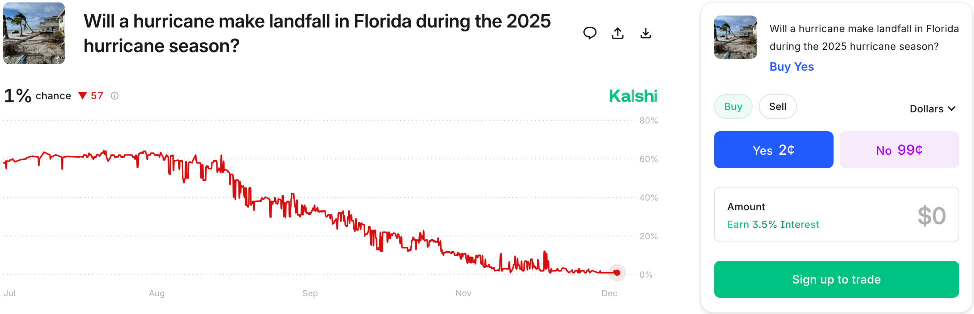


[5] [Xia, 2025] also uses the same trick.

Current Progress

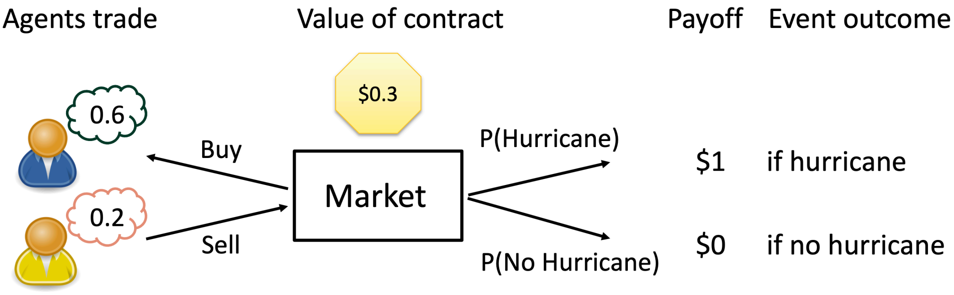
What is a prediction market?

- A prediction market is a financial market that is designed for information aggregation and prediction.
- Agents can "bet on beliefs", by trading contracts whose payoffs (e.g., binary payoff $\phi_w : \Omega \to \{0, 1\}$) are associated with an observed outcome in the future, $w \in \Omega$.



How do prediction markets aggregate information?

- Price ≈ Expectation of r.v. given all information



- Equilibrium price ≈ Value of contract ≈ Pr[Event | All information]

Other forecasting methods vs. prediction market

Opinion Poll

- Sample with equally weighted inputs
- No incentive to be truthful
- Hard to be real-time

Ask Experts

- Need to identify experts
- Hard to combine information

Machine Learning

- Need historical data, assuming past and future are related
- Hard to incorporate new information

Prediction Market

- Self-selection with bet-weighted inputs
- Monetary incentive
- No need for (assumptions on) data
- Real-time with new information immediately incorporated

Current Progress

Financial market vs. prediction market

Financial market

- Primary: capital allocation and hedge risk
- Secondary: information aggregation

Prediction market

- Primary: information aggregation
- Secondary: hedge risk

The goals are typically mixed together.

Risk and decision making under uncertainty

- Outcomes are in money ($): the r.v. x represents money (wealth or payoff).
- Utility of money $u(x)$: the utility an agent derives from that amount of money.

Risk and decision making under uncertainty

- Outcomes are in money (\$): the r.v. x represents money (wealth or payoff).
- Utility of money $u(x)$: the utility an agent derives from that amount of money.
- Risk attitudes
  - Risk neutral: $u(x) \sim x$
  - Risk averse (typical w/ diminishing marginal utility): u is concave, e.g., $u(x) \sim \log(x)$
  - Risk seeking: u is convex

Risk and decision making under uncertainty

- Outcomes are in money ($): the r.v. x represents money (wealth or payoff).
- Utility of money $u(x)$: the utility an agent derives from that amount of money.
- Risk attitudes
  - Risk neutral: $u(x) \sim x$
  - Risk averse (typical w/ diminishing marginal utility): u is concave, e.g., $u(x) \sim \log(x)$
  - Risk seeking: u is convex
- Absolute risk aversion: $r_u(x) = -u''(x)/u'(x)$
  - The larger the number, the more the agent is risk averse.

Risk and decision making under uncertainty

- Outcomes are in money ($): the r.v. x represents money (wealth or payoff).
- Utility of money $u(x)$: the utility an agent derives from that amount of money.
- Risk attitudes
  - Risk neutral: $u(x) \sim x$
  - Risk averse (typical w/ diminishing marginal utility): u is concave, e.g., $u(x) \sim \log(x)$
  - Risk seeking: u is convex
- Absolute risk aversion: $r_u(x) = -u''(x)/u'(x)$
  - The larger the number, the more the agent is risk averse.
- Expected utility: $\sum_w Pr(w)u(x_w)$

Risk attitude, hedging, and risk allocation [6]

Example:

- I'm risk averse w/ $u(x) = \log(x)$; the insurance company is risk neutral w/ $u(x) = x$.
- I believe that my car might be destroyed by a hurricane with prob. 0.01.
- $\Omega = \{w_1, w_2\}$. w_1: car destroyed. w_2: car not destroyed.

[6] Example adapted from Yiling Chen's slides.

Risk attitude, hedging, and risk allocation [6]

Example:

- I'm risk averse w/ $u(x) = \log(x)$; the insurance company is risk neutral w/ $u(x) = x$.
- I believe that my car might be destroyed by a hurricane with prob. 0.01.
- $\Omega = \{w_1, w_2\}$. w_1: car destroyed. w_2: car not destroyed.

- Suppose $u(w_1) = \log(10,000)$ and $u(w_2) = \log(20,000)$.
  $\mathbb{E}[u] = 0.01 \cdot \log(10,000) + 0.99 \cdot \log(20,000)$

[6] Example adapted from Yiling Chen's slides.

Risk attitude, hedging, and risk allocation [6]

Example:

- I'm risk averse w/ $u(x) = \log(x)$; the insurance company is risk neutral w/ $u(x) = x$.
- I believe that my car might be destroyed by a hurricane with prob. 0.01.
- $\Omega = \{w_1, w_2\}$. w_1: car destroyed. w_2: car not destroyed.

- Suppose $u(w_1) = \log(10,000)$ and $u(w_2) = \log(20,000)$.
  $\mathbb{E}[u] = 0.01 \cdot \log(10,000) + 0.99 \cdot \log(20,000)$
- I will buy \$10,000 insurance for \$125
  $\mathbb{E}[u_{buy}] = 0.01 \cdot \log(19,875) + 0.99 \cdot \log(19,875) > \mathbb{E}[u]$

[6] Example adapted from Yiling Chen's slides.

Risk attitude, hedging, and risk allocation [6]

Example:

- I'm risk averse w/ $u(x) = \log(x)$; the insurance company is risk neutral w/ $u(x) = x$.
- I believe that my car might be destroyed by a hurricane with prob. 0.01.
- $\Omega = \{w_1, w_2\}$. w_1: car destroyed. w_2: car not destroyed.

- Suppose $u(w_1) = \log(10,000)$ and $u(w_2) = \log(20,000)$.
  $\mathbb{E}[u] = 0.01 \cdot \log(10,000) + 0.99 \cdot \log(20,000)$
- I will buy \$10,000 insurance for \$125
  $\mathbb{E}[u_{buy}] = 0.01 \cdot \log(19,875) + 0.99 \cdot \log(19,875) > \mathbb{E}[u]$

- Suppose that the insurance company also believes $Pr(\text{car destroyed}) = 0.01$
  $\mathbb{E}[u_{ins}] = 0.01 \cdot (-9,875) + 0.99 \cdot (125) > 0$

[6] Example adapted from Yiling Chen's slides.

Risk attitude, hedging, and risk allocation [6]

Example:

- I'm risk averse w/ $u(x) = \log(x)$; the insurance company is risk neutral w/ $u(x) = x$.
- I believe that my car might be destroyed by a hurricane with prob. 0.01.
- $\Omega = \{w_1, w_2\}$. w_1: car destroyed. w_2: car not destroyed.

- Suppose $u(w_1) = \log(10,000)$ and $u(w_2) = \log(20,000)$.
  $\mathbb{E}[u] = 0.01 \cdot \log(10,000) + 0.99 \cdot \log(20,000)$
- I will buy \$10,000 insurance for \$125
  $\mathbb{E}[u_{buy}] = 0.01 \cdot \log(19,875) + 0.99 \cdot \log(19,875) > \mathbb{E}[u]$

- Suppose that the insurance company also believes $Pr(\text{car destroyed}) = 0.01$
  $\mathbb{E}[u_{ins}] = 0.01 \cdot (-9,875) + 0.99 \cdot (125) > 0$

The transaction allocates risk. Everyone is happy.

[6] Example adapted from Yiling Chen's slides.

Probability and speculating [7]

Example (continued):

- Suppose that I'm risk neutral $u(x) = x$, and believe that $Pr(\text{car destroyed}) = 0.02$.
- I will buy \$10,000 insurance for \$125
  The insurance is a contract: \$10,000 if car destroyed, 0 otherwise.
  $\mathbb{E}[\text{Insurance}] = 0.02 \cdot (10,000) + 0.98 \cdot (0) > \$125$
- I get \$75 on expectation.

[7]Example adapted from Yiling Chen's slides.

Probability and speculating [7]

Example (continued):

- Suppose that I'm risk neutral $u(x) = x$, and believe that $Pr(\text{car destroyed}) = 0.02$.
- I will buy \$10,000 insurance for \$125
  The insurance is a contract: \$10,000 if car destroyed, 0 otherwise.
  $\mathbb{E}[\text{Insurance}] = 0.02 \cdot (10,000) + 0.98 \cdot (0) > \$125$
- I get \$75 on expectation.

The transaction speculates the insurance company.

[7] Example adapted from Yiling Chen's slides.

Probability and speculating [7]

Example (continued):

- Suppose that I'm risk neutral $u(x) = x$, and believe that $Pr(\text{car destroyed}) = 0.02$.
- I will buy \$10,000 insurance for \$125
  The insurance is a contract: \$10,000 if car destroyed, 0 otherwise.
  $\mathbb{E}[\text{Insurance}] = 0.02 \cdot (10,000) + 0.98 \cdot (0) > \$125$
- I get \$75 on expectation.

The transaction speculates the insurance company.

Prediction market generalize to

- arbitrary states;
- more than two parties.

[7] Example adapted from Yiling Chen's slides.

Probability and speculating [7]

Example (continued):

- Suppose that I'm risk neutral $u(x) = x$, and believe that $Pr(\text{car destroyed}) = 0.02$.
- I will buy \$10,000 insurance for \$125
  The insurance is a contract: \$10,000 if car destroyed, 0 otherwise.
  $\mathbb{E}[\text{Insurance}] = 0.02 \cdot (10,000) + 0.98 \cdot (0) > \$125$
- I get \$75 on expectation.

The transaction speculates the insurance company.

Prediction market generalize to

- arbitrary states;
- more than two parties.

Design market mechanisms to allow speculation and allocate risk among participants.

[7] Example adapted from Yiling Chen's slides.

Risk-neutral probability

- Subjective probability is an agent's personal judgment
  Can be mixed with the agent's utility (risk attitude)

Risk-neutral probability

- Subjective probability is an agent's personal judgment
  Can be mixed with the agent's utility (risk attitude)
- Risk-neutral probability: the probability that a risk-neutral agent has to have the same expected utility

$$\sum_w Pr^{rn}(w) x_w = \sum_w Pr(w) u(x_w)$$

Risk-neutral probability

- Subjective probability is an agent's personal judgment
  Can be mixed with the agent's utility (risk attitude)
- Risk-neutral probability: the probability that a risk-neutral agent has to have the same expected utility

$$\sum_w Pr^{rn}(w)x_w = \sum_w Pr(w)u(x_w)$$

- Risk neutral probability is the normalized product of subjective probability and marginal utility

$$\sum_w Pr^{rn}(w) \sim Pr(w)u'(x_w)$$

Market design: contracts

1. Random variable: turn an uncertain event of interest into a random variable
   - Binary, discrete: {win, lose}, {sunny, rainy, cloudy}
   - Continuous: temperature, price, time, vote share...
2. Payoff functions
   - Arrow-Debreu: \$1 if the event happens, and \$0 otherwise
   - Index / continuous: the payoff scales with the result
   - Other forms: dividends, pari-mutuel, options
3. Payoff output
   - Real money: USD, cryptocurrency
   - Play money: virtual points for fun, reputation, etc.
   - Other forms: prize, lottery, etc.

Market design: mechanisms

- Call market
  - *Mechanism:* Orders are collected into a "batch" over a period of time and then executed at once at a *single* clearing price that maximizes the volume of trade; There are different price determination rules.
  - *Applications:* Opening price, CoW Swap, illiquid asset markets.
  - *Characteristics:* Rely on counterparties, not real-time, alleviate thin market problem.

Market design: mechanisms

- Call market
  - *Mechanism:* Orders are collected into a "batch" over a period of time and then executed at once at a *single* clearing price that maximizes the volume of trade; There are different price determination rules.
  - *Applications:* Opening price, CoW Swap, illiquid asset markets.
  - *Characteristics:* Rely on counterparties, not real-time, alleviate thin market problem.
- Continuous double auction (CDA)
  - *Mechanism:* Buy and sell orders continuously come in and are aggregated in a central *limit order book* (CLOB) (i.e., call market w/ period$\rightarrow 0$);
    As bid $\geq$ ask, a transaction occurs at the incumbent order price.
  - *Applications:* Most financial markets.
  - *Characteristics:* Rely on counterparties, real time, may suffer thin market problem.

Market design: mechanisms

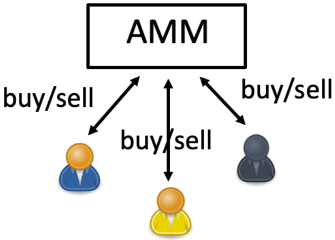
- Call market
  - *Mechanism:* Orders are collected into a "batch" over a period of time and then executed at once at a *single* clearing price that maximizes the volume of trade; There are different price determination rules.
  - *Applications:* Opening price, CoW Swap, illiquid asset markets.
  - *Characteristics:* Rely on counterparties, not real-time, alleviate thin market problem.

- Continuous double auction (CDA)
  - *Mechanism:* Buy and sell orders continuously come in and are aggregated in a central *limit order book* (CLOB) (i.e., call market w/ period$\to$ 0); As bid $\geq$ ask, a transaction occurs at the incumbent order price.
  - *Applications:* Most financial markets.
  - *Characteristics:* Rely on counterparties, real time, may suffer thin market problem.

- Automated market maker (AMM)
  - Always willing to quote prices and offer to trade any quantity.
  - No need for counterparties, real time, improve liquidity (thus information aggregation).

Automated market maker (AMM)

- Always offer to buy or sell at some price;
  How to decide the prices?
- If shares are bought, increase the price (i.e., reflect the market belief);
  How to update the prices?
- May subsidize the market for information.
  Can we leverage proper scoring rules?



Current prediction market landscape

Decentralized (blockchain-based)

Characteristics: Global access, non-custodial, crypto settlement (USDC, SOL, etc.).

- **Polymarket** (Polygon)
  - *Status:* Global volume leader.
  - *Mech:* Hybrid CLOB (off-chain matching, on-chain settlement).
- **Drift Protocol** (Solana)
  - *Status:* Leading Solana Market.
  - *Mech:* Hybrid CLOB with cross-collateral.
- **Limitless** (Base)
  - *Status:* Leader on Coinbase's L2.
  - *Mech:* On-chain CLOB (short-term focus).
- **Azuro** (Gnosis/Polygon)
  - *Mech:* Liquidity pool / AMM (peer-to-pool).

Current prediction market landscape

Centralized & Regulated (US focused)

Characteristics: KYC required, bank transfers (USD), legal compliance.

- **Kalshi** (CFTC Regulated)
  - *Status:* US market leader.
  - *Mech:* Centralized exchange.
- **Fanatics Markets** (acquired Paragon Global Markets, LLC)
  - *Status:* New entrant (2025).
  - *Mech:* Consumer app backed by Crypto.com exchange.
- **PredictIt**
  - *Status:* Legacy / academic, not for profit.
  - *Mech:* Low limits, No-Action letter (2014-2022).

Alternative Model

- **Manifold**
  - *Mech:* Play money (Mana) & redeemable cash (Sweepcash).

Current Progress

Incentives for trading: Leveraging scoring rules

	1 person	$n > 1$ people
Elicit belief (verification)	scoring rule	prediction market
Elicit signal (no verification)	x	peer prediction

Incentives for trading: Leveraging scoring rules

	1 person	$n > 1$ people
Elicit belief (verification)	scoring rule	prediction market
Elicit signal (no verification)	x	peer prediction

Recap:

Definition (Strictly proper scoring rule)

A scoring rule S is *strictly proper* if for all $\hat{\boldsymbol{p}} \neq \boldsymbol{p} \in \Delta_\Omega$,

$$S(\boldsymbol{p}, \boldsymbol{p}) > S(\hat{\boldsymbol{p}}, \boldsymbol{p})$$

where $S(\hat{\boldsymbol{p}}, \boldsymbol{p}) := \mathbb{E}_{w \sim \boldsymbol{p}}[S(\hat{\boldsymbol{p}}, w)]$.

Incentives for trading: Leveraging scoring rules

Myopic incentives: optimal to trade until instantaneous price $\pi = \boldsymbol{p}$ (agent belief)

Connect to *sequential* proper scoring rule

- Consider outcome space $w \in \Omega = \{yes, no\}$
- Initialize the market report: $\hat{\boldsymbol{p}}^{(0)}$ is uniform;
- Receive sequence of reports from agent 1 to n: $\hat{\boldsymbol{p}}^{(1)}, \hat{\boldsymbol{p}}^{(2)}, \ldots, \hat{\boldsymbol{p}}^{(n)}$;
- Upon *realization* of w_k, the i-th agent pays

$$S(\hat{\boldsymbol{p}}^{(i-1)}, w_k) - S(\hat{\boldsymbol{p}}^{(i)}, w_k);$$

Incentives for trading: Leveraging scoring rules

Myopic incentives: optimal to trade until instantaneous price $\pi = \boldsymbol{p}$ (agent belief)

Connect to *sequential* proper scoring rule

- Consider outcome space $w \in \Omega = \{yes, no\}$
- Initialize the market report: $\hat{\boldsymbol{p}}^{(0)}$ is uniform;
- Receive sequence of reports from agent 1 to n: $\hat{\boldsymbol{p}}^{(1)}, \hat{\boldsymbol{p}}^{(2)}, \ldots, \hat{\boldsymbol{p}}^{(n)}$;
- Upon *realization* of w_k, the i-th agent pays

$$S(\hat{\boldsymbol{p}}^{(i-1)}, w_k) - S(\hat{\boldsymbol{p}}^{(i)}, w_k);$$

- Take S to be any strictly proper scoring rule, it is rational to report truthfully in position i, $\hat{\boldsymbol{p}}^{(i)} = \boldsymbol{p}^{(i)}$, i.e., minimizing payment.

Incentives for trading: Leveraging scoring rules

Myopic incentives: optimal to trade until instantaneous price $\pi = \boldsymbol{p}$ (agent belief)

Connect to *sequential* proper scoring rule

- Consider outcome space $w \in \Omega = \{yes, no\}$
- Initialize the market report: $\hat{\boldsymbol{p}}^{(0)}$ is uniform;
- Receive sequence of reports from agent 1 to n: $\hat{\boldsymbol{p}}^{(1)}, \hat{\boldsymbol{p}}^{(2)}, \ldots, \hat{\boldsymbol{p}}^{(n)}$;
- Upon *realization* of w_k, the i-th agent pays

$$S(\hat{\boldsymbol{p}}^{(i-1)}, w_k) - S(\hat{\boldsymbol{p}}^{(i)}, w_k);$$

- Take S to be any strictly proper scoring rule, it is rational to report truthfully in position i, $\hat{\boldsymbol{p}}^{(i)} = \boldsymbol{p}^{(i)}$, i.e., minimizing payment.
- The cost to market designer (w/ uniform prior)

$$S(\hat{\boldsymbol{p}}^{(n)}, w_k) - S(\hat{\boldsymbol{p}}^{(0)}, w_k) \leq b\ln(1) - b\ln(1/n) = b\ln(n).$$

Market scoring rules [Hanson, 2003, Hanson, 2007]

- Use a proper scoring rule;
- A trader can change the current probability estimate to a new one;
- The trader pays (receives) the scoring rule payment according to the old probability estimate and the outcome.

Incentives for trading: Leveraging scoring rules

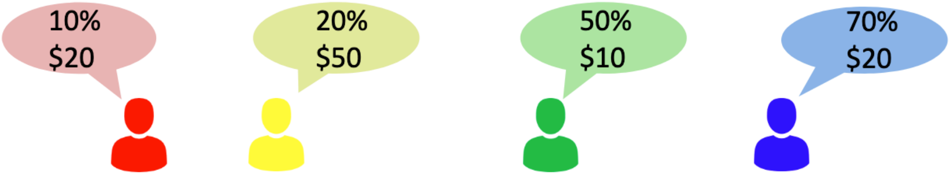
Wagering mechanisms

- Each agent reports a forecast $\hat{\boldsymbol{p}}_i$ and a wager $\boldsymbol{\delta}_i$;
- The mechanism redistributes wagers upon realization of $w \in \Omega$.

Wagering mechanisms

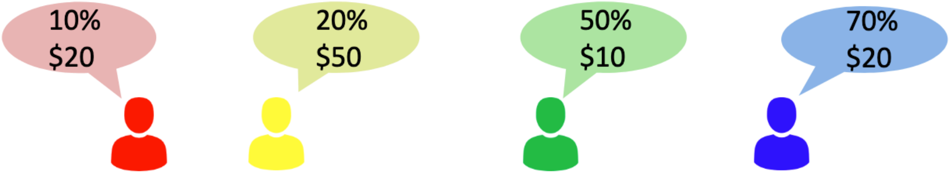
- Each agent reports a forecast $\hat{\boldsymbol{p}}_i$ and a wager $\boldsymbol{\delta}_i$;
- The mechanism redistributes wagers upon realization of $w \in \Omega$.

- Example: Will S&P price increase tomorrow?



Wagering mechanisms

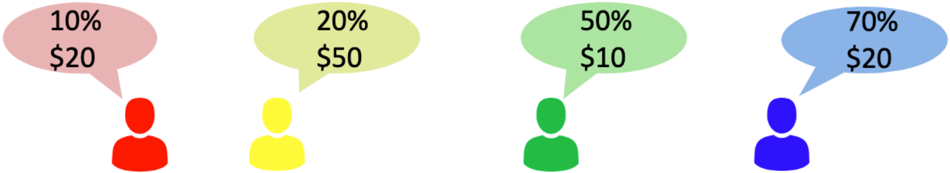
- Each agent reports a forecast $\hat{\boldsymbol{p}}_i$ and a wager $\boldsymbol{\delta}_i$;
- The mechanism redistributes wagers upon realization of $w \in \Omega$.
  According to scoring rules!
- Example: Will S&P price increase tomorrow?



Incentives for trading: Leveraging scoring rules

Wagering mechanisms

- Each agent reports a forecast $\hat{\boldsymbol{p}}_i$ and a wager $\boldsymbol{\delta}_i$;
- The mechanism redistributes wagers upon realization of $w \in \Omega$.
  According to scoring rules!
- Example: Will S&P price increase tomorrow?

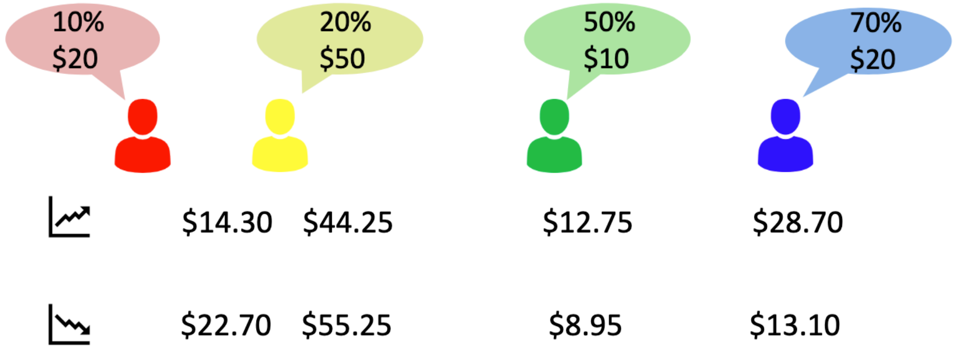


- Weighted-score wagering mechanism [Lambert et al., 2015]

$$\pi_i(\boldsymbol{p}, \boldsymbol{\delta}, w) = \delta_i \left(1 + S(p_i, w) - \frac{\sum_{j \neq i} \delta_j S(p_j, w)}{\sum_{j \neq i} \delta_j} \right), \quad r_i(\boldsymbol{p}, \boldsymbol{\delta}, w) = \pi_i(\boldsymbol{p}, \boldsymbol{\delta}, w) - \delta_i$$

Wagering mechanisms

- Each agent reports a forecast $\hat{\boldsymbol{p}}_i$ and a wager $\boldsymbol{\delta}_i$;
- The mechanism redistributes wagers upon realization of $w \in \Omega$.
  According to scoring rules! Example: Will S&P price increase tomorrow?



Cost-function-based AMM

Assume outcome space of n possible outcomes.

Cost-function-based AMMs

- Maintain the market state, $\boldsymbol{q} = (q_1, ..., q_n)$, i.e., shares sold for each security (outcome i);

	Yes	No
Initialization	0	0
Buy 2 for Yes	2	0
Buy 5 for Yes	7	0
Buy 2 for No	7	2
Sell 1 for Yes	6	2

Cost-function-based AMM

Assume outcome space of n possible outcomes.

Cost-function-based AMMs

- Maintain the market state, $\boldsymbol{q} = (q_1, ..., q_n)$, i.e., shares sold for each security (outcome i);
- Use a convex, differentiable cost function $C : \mathbb{R}^{|\Omega|} \to \mathbb{R}$;

	Yes	No
Initialization	0	0
Buy 2 for Yes	2	0
Buy 5 for Yes	7	0
Buy 2 for No	7	2
Sell 1 for Yes	6	2

Cost-function-based AMM

Assume outcome space of n possible outcomes.

Cost-function-based AMMs

- Maintain the market state, $\boldsymbol{q} = (q_1, ..., q_n)$, i.e., shares sold for each security (outcome i);
- Use a convex, differentiable cost function $C : \mathbb{R}^{|\Omega|} \to \mathbb{R}$;
- Quote instantaneous price as

$$p_i(\boldsymbol{q}) = \partial C(\boldsymbol{q}) / \partial q_i;$$

	Yes	No
Initialization	0	0
Buy 2 for Yes	2	0
Buy 5 for Yes	7	0
Buy 2 for No	7	2
Sell 1 for Yes	6	2

Cost-function-based AMM

Assume outcome space of n possible outcomes.

Cost-function-based AMMs

- Maintain the market state, $\boldsymbol{q} = (q_1, ..., q_n)$, i.e., shares sold for each security (outcome i);
- Use a convex, differentiable cost function $C : \mathbb{R}^{|\Omega|} \rightarrow \mathbb{R}$;
- Quote instantaneous price as

$$p_i(\boldsymbol{q}) = \partial C(\boldsymbol{q}) / \partial q_i;$$

- Charge a trader who buys a bundle $\boldsymbol{\delta} \in \mathbb{R}^{|\Omega|}$ of contracts by $C(\boldsymbol{q} + \boldsymbol{\delta}) - C(\boldsymbol{q})$;

	Yes	No
Initialization	0	0
Buy 2 for Yes	2	0
Buy 5 for Yes	7	0
Buy 2 for No	7	2
Sell 1 for Yes	6	2

Cost-function-based AMM

Assume outcome space of n possible outcomes.

Cost-function-based AMMs

- Maintain the market state, $\boldsymbol{q} = (q_1, ..., q_n)$, i.e., shares sold for each security (outcome i);
- Use a convex, differentiable cost function $C : \mathbb{R}^{|\Omega|} \to \mathbb{R}$;
- Quote instantaneous price as

$$p_i(\boldsymbol{q}) = \partial C(\boldsymbol{q}) / \partial q_i;$$

- Charge a trader who buys a bundle $\boldsymbol{\delta} \in \mathbb{R}^{|\Omega|}$ of contracts by $C(\boldsymbol{q} + \boldsymbol{\delta}) - C(\boldsymbol{q})$;
- Update market state after each trade: $\boldsymbol{q} \leftarrow \boldsymbol{q} + \boldsymbol{\delta}$.

	Yes	No
Initialization	0	0
Buy 2 for Yes	2	0
Buy 5 for Yes	7	0
Buy 2 for No	7	2
Sell 1 for Yes	6	2

Some desirable properties for AMMs

- No "round-trip" arbitrage
- Prices nonnegative, sum to one (i.e., probability)
- Responsiveness
- Liquidity (i.e., relatively small price change after a small trade)
- Bounded budget or loss to AMM
- Individual rationality
- Expressiveness (i.e., allow traders to bet on any possible outcome)
- Computational complexity

Logarithmic market scoring rule (LMSR)

Logarithmic market scoring rule (LMSR) AMMs

- Use cost functions:
$$C(\boldsymbol{q}) = b \log\Big(\sum_i e^{q_i/b}\Big),$$
where b is called the liquidity parameter;
- Quote instantaneous prices:
$$p_i(\boldsymbol{q}) = \frac{e^{q_i/b}}{\sum_j e^{q_j/b}};$$
- Charge a trader who buys a bundle $\boldsymbol{\delta} \in \mathbb{R}^{|\Omega|}$ of contracts by $C(\boldsymbol{q} + \boldsymbol{\delta}) - C(\boldsymbol{q})$;
- Update market state after each trade: $\boldsymbol{q} \leftarrow \boldsymbol{q} + \boldsymbol{\delta}$.

Some desirable properties for AMMs

LMSR AMMs satisfy

- No “round-trip” arbitrage
- Prices nonnegative, sum to one (i.e., probability)
- Responsiveness
- Liquidity (i.e., relatively small price change after a small trade)
- Bounded budget or loss to AMM
- Individual rationality

Example: LMSR AMM

A prediction market: *Will a hurricane make landfall in Florida in 2026?*

Assume an LMSR AMM with $b = 1$, so $C(\boldsymbol{q}) = \ln(e^{q_0} + e^{q_1})$ and $S(\boldsymbol{p}, w_i) = \ln(p_i)$

	Yes	**No**	**Payment**	$\pi(Yes)$	$\pi(No)$	**Payment | Yes**	**Payment | No**
Initialization	0	0	–	0.5	0.5	–	–
Buy 1 for Yes	1	0	0.62 $\ln(e^1 + e^0)$ $-\ln(e^0 + e^0)$	0.73 $e^1/(e^1 + e^0)$	0.27	-0.38 $\ln(0.5)-$ $\ln(0.73)$	0.62 $\ln(0.5)-$ $\ln(0.27)$
Buy 2 for Yes	3	0	1.73 $\ln(e^3 + e^0)$ $-\ln(e^1 + e^0)$	0.95 $e^3/(e^3 + e^0)$	0.05	-0.26 $\ln(0.73)-$ $\ln(0.95)$	1.73 $\ln(0.27)-$ $\ln(0.05)$
Buy 1 for No	3	1	0.08 $\ln(e^3 + e^1)$ $-\ln(e^3 + e^0)$	0.88 $e^3/(e^3 + e^1)$	0.12	0.08 $\ln(0.95)-$ $\ln(0.88)$	-0.92 $\ln(0.05)-$ $\ln(0.12)$

Other market scoring rule AMMs

Quadratic market scoring rule (QMSR) AMMs (derived from the Brier scoring rule)

- Use cost functions:

$$C(\boldsymbol{q}) = \frac{\sum_{i=1}^{n} q_i}{n} + \frac{\sum_{i=1}^{n} q_i^2}{4b} - \frac{\left(\sum_{i=1}^{n} q_i\right)^2}{4bn} - \frac{b}{n},$$

  where $b > 0$ is the liquidity parameter.

- Quote instantaneous prices:

$$p_i(\boldsymbol{q}) = \frac{1}{n} + \frac{q_i}{2b} - \frac{\sum_{j=1}^{n} q_j}{2nb}$$

Other market scoring rule AMMs: Decentralized exchange

Constant function market maker (CFMM) for n assets maintains

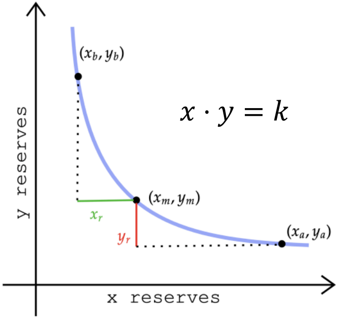
- A reserve of available assets $\boldsymbol{q} \in \mathbb{R}^n$;
- A trading function $\phi : \mathbb{R}^n \to \mathbb{R}$ that is concave and increasing;
- A trade or swap $\boldsymbol{\delta} \in \mathbb{R}^n$ following $\phi(\boldsymbol{q} + \boldsymbol{\delta}) = \phi(\boldsymbol{q})$.

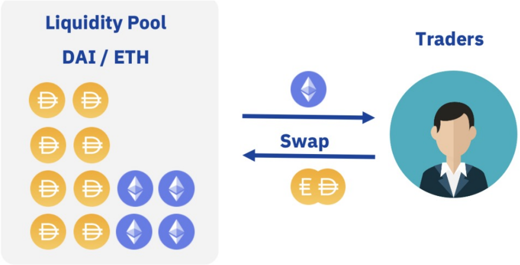
Other market scoring rule AMMs: Decentralized exchange

Constant function market maker (CFMM) for n assets maintains

- A reserve of available assets $\boldsymbol{q} \in \mathbb{R}^n$;
- A trading function $\phi : \mathbb{R}^n \to \mathbb{R}$ that is concave and increasing;
- A trade or swap $\boldsymbol{\delta} \in \mathbb{R}^n$ following $\phi(\boldsymbol{q} + \boldsymbol{\delta}) = \phi(\boldsymbol{q})$.

Example: Constant product market maker (CPMM) employed by Uniswap, Balancer, etc.





CFMMs $\Leftrightarrow$ prediction markets

CFMMs

- Trades: assets $\leftrightarrow$ assets
- AMM: providing liquidity & facilitating swaps

Prediction markets

- Trades: securities $\leftrightarrow$ cash
- AMM: information elicitation & aggregation

CFMMs $\Leftrightarrow$ prediction markets

CFMMs

- Trades: assets $\leftrightarrow$ assets
- AMM: providing liquidity & facilitating swaps

Prediction markets

- Trades: securities $\leftrightarrow$ cash
- AMM: information elicitation & aggregation

Theorem [Frongillo et al., 2024]

CFMMs and **Cost-function market makers** are **equivalent** (i.e. have same available trades for a given history), via following maps:

$$\psi_1 : \phi \mapsto C, \text{ where } C(\mathsf{q}) := \inf\{c \in \mathbb{R} \mid \phi(c \cdot 1 - \mathsf{q}) \geq \phi(\mathsf{q}_0)\}$$
$$\psi_2 : C \mapsto \phi, \text{ where } \phi(\mathsf{q}) := -C(-\mathsf{q}) \; .$$

CFMMs ⇔ prediction markets

CFMMs

- Trades: assets $\leftrightarrow$ assets
- AMM: providing liquidity & facilitating swaps

Prediction markets

- Trades: securities $\leftrightarrow$ cash
- AMM: information elicitation & aggregation

Theorem [Frongillo et al., 2024]

CFMMs and **Cost-function market makers** are **equivalent** (i.e. have same available trades for a given history), via following maps:

$$\psi_1 : \phi \mapsto C, \text{ where } C(\mathsf{q}) := \inf\{c \in \mathbb{R} \mid \phi(c \cdot 1 - \mathsf{q}) \geq \phi(\mathsf{q}_0)\}$$
$$\psi_2 : C \mapsto \phi, \text{ where } \phi(\mathsf{q}) := -C(-\mathsf{q}) \ .$$

Intuition: A prediction market of n securities is a market of $n+1$ assets (securities & cash). A *cashless prediction market* replaces any \$1 cash payment with one of each security / asset, which is a CFMM.

Example: CPMMs $\Leftrightarrow$ cost-function AMM

- The cost function equivalent of CPMM (i.e., $\phi(\boldsymbol{q}) = \sqrt{q_1 \cdot q_2} = k$) is

$$C_k(\mathrm{q}) = -k + \frac{1}{2}\left(q_1 + q_2 + \sqrt{4k^2 + (q_1 - q_2)^2}\right);$$

Example: CPMMs ⇔ cost-function AMM

- The cost function equivalent of CPMM (i.e., $\phi(\boldsymbol{q}) = \sqrt{q_1 \cdot q_2} = k$) is

$$C_k(\mathrm{q}) = -k + \frac{1}{2}\left(q_1 + q_2 + \sqrt{4k^2 + (q_1 - q_2)^2}\right);$$

- The cost function is also the implicit function of a constant-log-utility market maker [Chen and Pennock, 2012] with utility function, $u(x) = \log(k + x)$ with $k > 0$;

Example: CPMMs $\Leftrightarrow$ cost-function AMM

- The cost function equivalent of CPMM (i.e., $\phi(\boldsymbol{q}) = \sqrt{q_1 \cdot q_2} = k$) is

$$C_k(\mathrm{q}) = -k + \frac{1}{2}\left(q_1 + q_2 + \sqrt{4k^2 + (q_1 - q_2)^2}\right);$$

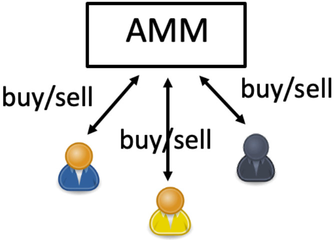
- The cost function is also the implicit function of a constant-log-utility market maker [Chen and Pennock, 2012] with utility function, $u(x) = \log(k + x)$ with $k > 0$;
- The corresponding proper scoring rule is

$$S_k(\mathrm{p}, w_i) = -k\sqrt{\frac{1 - p_i}{p_i}}$$

Boosting loss scoring rule [Buja et al., 2005].

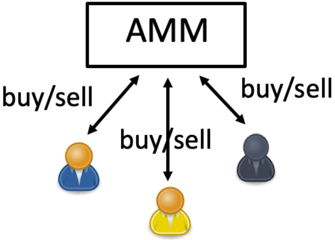
Design AMMs as online algorithms

- Given a set of outcomes Ω, a cost function C, and initial market state $\boldsymbol{q}^{(0)}$, in each round t:
  1. Price(i): return price of outcome i, i.e., $p_i(\boldsymbol{q}^{(t)})$;
  2. Cost(i, s) for $s \in \mathbb{R}$: return the cost of buying s shares of outcome i, i.e., $C(\boldsymbol{q}^{(t)} + s \cdot 1_i) - C(\boldsymbol{q}^{(t)})$;
  3. Buy(i, s): charge Cost(i, s) and update $\boldsymbol{q}^{(t+1)} \leftarrow \boldsymbol{q}^{(t)} + s \cdot 1_i$



Design AMMs as online algorithms

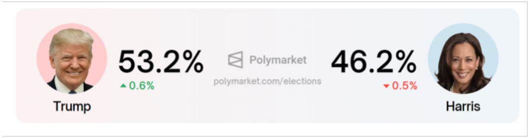
- Given a set of outcomes Ω, a cost function C, and initial market state $\boldsymbol{q}^{(0)}$, in each round t:
  1. Price(i): return price of outcome i, i.e., $p_i(\boldsymbol{q}^{(t)})$;
  2. Cost(i, s) for $s \in \mathbb{R}$: return the cost of buying s shares of outcome i, i.e., $C(\boldsymbol{q}^{(t)} + s \cdot 1_i) - C(\boldsymbol{q}^{(t)})$;
  3. Buy(i, s): charge Cost(i, s) and update $\boldsymbol{q}^{(t+1)} \leftarrow \boldsymbol{q}^{(t)} + s \cdot 1_i$



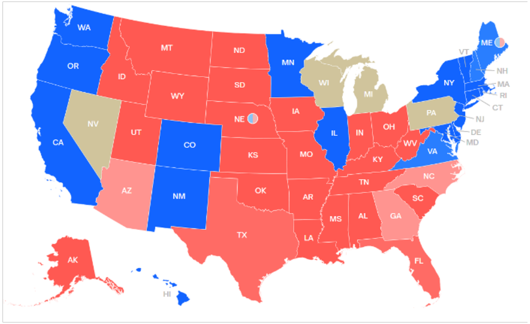
Goal: Design an algorithm or data structure to support above market operation efficiently.

Current Progress

Prediction market: from binary to large outcome space

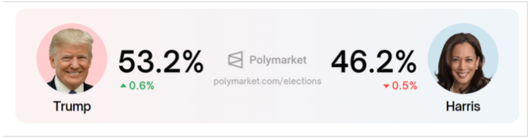


Ω ={Trump wins, Harris wins}; $|\Omega| = 2$.

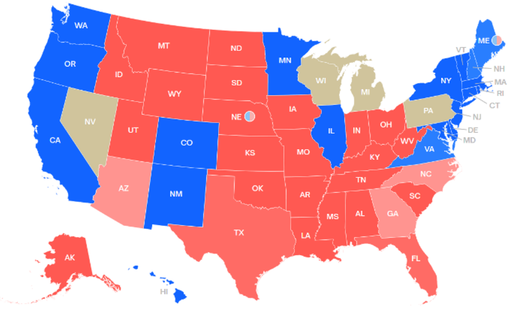


Ω ={Each state's winner}; $|\Omega| = 2^{50}$.

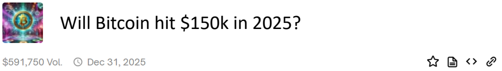
Prediction market: from binary to large outcome space



$\Omega = $ {Trump wins, Harris wins}; $|\Omega| = 2.$



$\Omega = $ {Each state's winner}; $|\Omega| = 2^{50}.$



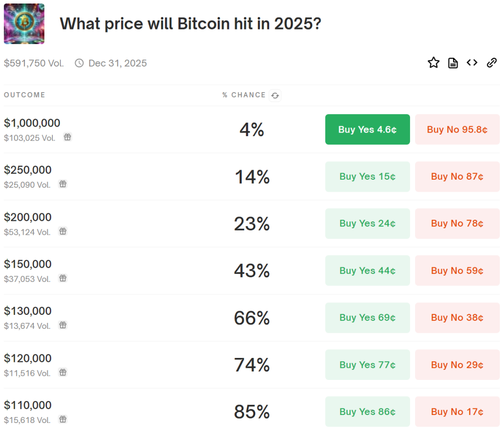
$\Omega = $ {Yes, No}; $|\Omega| = 2.$

$$\Omega = \mathbb{R}; \ |\Omega| = \infty.$$

Traditional market implementation

- Predetermined discretizations, independent markets

  May suffer
  - Thin market problem
  - Logic inconsistency
  - Arbitrage opportunities



Traditional market implementation

- Predetermined discretizations, independent markets

  May suffer
  - Thin market problem
  - Logic inconsistency
  - Arbitrage opportunities

- How about some combinatorial prediction market for large Ω?

  May need to balance
  - Expressivenss
  - Computational complexity
  - Worst-case loss / liquidity

What price will Bitcoin hit in 2025?

$591,750 Vol. Dec 31, 2025

OUTCOME	% CHANCE		
$1,000,000 $103,025 Vol.	4%	Buy Yes 4.6¢	Buy No 95.8¢
$250,000 $25,090 Vol.	14%	Buy Yes 15¢	Buy No 87¢
$200,000 $53,124 Vol.	23%	Buy Yes 24¢	Buy No 78¢
$150,000 $37,053 Vol.	43%	Buy Yes 44¢	Buy No 59¢
$130,000 $13,674 Vol.	66%	Buy Yes 69¢	Buy No 38¢
$120,000 $11,516 Vol.	74%	Buy Yes 77¢	Buy No 29¢
$110,000 $15,618 Vol.	85%	Buy Yes 86¢	Buy No 17¢

Combinatorial prediction market

- Ω: A large outcome space with $n = |\Omega|$ possible outcomes;

  Example:

  1. $w_0 =$ FL: Democrats & PA: Democrats
  2. $w_1 =$ FL: Democrats & PA: Republicans
  3. $w_2 =$ FL: Republicans & PA: Democrats
  4. $w_3 =$ FL: Republicans & PA: Republicans
- $\mathcal{F} \subseteq 2^{\Omega}$: A set system that is a collection of subsets of Ω.

Combinatorial prediction market

- Ω: A large outcome space with $n = |\Omega|$ possible outcomes;

  Example:
  1. $w_0 =$ FL: Democrats & PA: Democrats
  2. $w_1 =$ FL: Democrats & PA: Republicans
  3. $w_2 =$ FL: Republicans & PA: Democrats
  4. $w_3 =$ FL: Republicans & PA: Republicans
- $\mathcal{F} \subseteq 2^{\Omega}$: A set system that is a collection of subsets of Ω.

- A prediction market $(\Omega, \mathcal{F})$ offers combinatorial security that
  - Specifies an event $E \in \mathcal{F}$;
    Example: "Republicans win Pennsylvania" (i.e., $E = \{w_1, w_3\}$), "The state outcomes differ" (i.e., $E = \{w_2, w_3\}$).
  - Pays \$1 if the event E happens.

Combinatorial prediction market

- Ω: A large outcome space with $n = |\Omega|$ possible outcomes;

  Example:
  1. w_0 = FL: Democrats & PA: Democrats
  2. w_1 = FL: Democrats & PA: Republicans
  3. w_2 = FL: Republicans & PA: Democrats
  4. w_3 = FL: Republicans & PA: Republicans
- $\mathcal{F} \subseteq 2^{\Omega}$: A set system that is a collection of subsets of Ω.

- A prediction market $(\Omega, \mathcal{F})$ offers combinatorial security that
  - Specifies an event $E \in \mathcal{F}$;
    Example: "Republicans win Pennsylvania" (i.e., $E = \{w_1, w_3\}$), "The state outcomes differ" (i.e., $E = \{w_2, w_3\}$).
  - Pays \$1 if the event E happens.

- Examples of popular set systems.

Example: Interval Security

- $\mathcal{F}$: A collection of all intervals [Dudík et al., 2021]
- A prediction market $(\Omega, \mathcal{F})$ offers combinatorial security that specifies
  - An interval
  - Pays \$1 if the outcome falls in the interval
  - Expressiveness: precision level

Q1, 2021 (or before)	1¢
Q2, 2021	27¢
Q3, 2021	55¢
Q4, 2021 (or later)	17¢

Example: Interval Security

- $\mathcal{F}$: A collection of all intervals [Dudík et al., 2021]
- A prediction market $(\Omega, \mathcal{F})$ offers combinatorial security that specifies
  - An interval
  - Pays \$1 if the outcome falls in the interval
  - Expressiveness: precision level
- d-dimensional orthogonal security
  Example: "NVDA $\in [180, 190)$ & GOOGL $\in [320, 330)$"

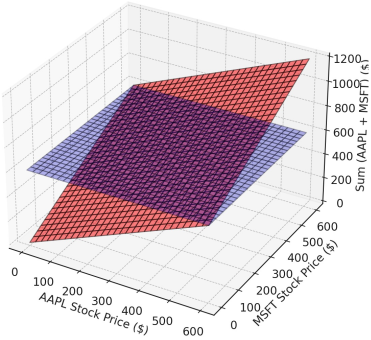
Q1, 2021 (or before)	1¢
Q2, 2021	27¢
Q3, 2021	55¢
Q4, 2021 (or later)	17¢

Example: Hyperplane Security

- $\Omega \subset \mathbb{R}^d$
- $\mathcal{F}$: A collection of half-space associated with hyperplanes [Wang et al., 2021]

$$E_{\beta,\beta_0} = \{w \in \Omega : \beta^T w + \beta_0 \geq 0\}$$

- A prediction market $(\Omega, \mathcal{F})$ offers combinatorial security that specifies
  - A half-space
  - Pays \$1 if the outcome falls in the half-space



APPL+MSFT $\geq$ 600

Example: Top L Candidates

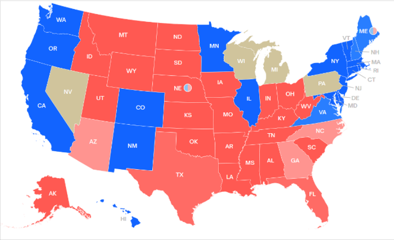
- $\mathcal{F}$: A subset of L candidates among K candidates
- A prediction market $(\Omega, \mathcal{F})$ offers combinatorial security that specifies
  - A set of L candidates
  - Pays \$1 if the top L candidates are from the set

Example: Permutations

- $\mathcal{F}$: A collection of pair comparisons among K candidates [Chen et al., 2007]
- A prediction market $(\Omega, \mathcal{F})$ offers combinatorial security that specifies
  - A pair (a, b) where candidate a ranks higher than candidate b
  - Pays \$1 if the pair comparison turns out to be true

Example: Boolean Betting

- $\mathcal{F}$: Any conjunction of event outcomes [Chen et al., 2008]
- A prediction market $(\Omega, \mathcal{F})$ offers combinatorial security that specifies
  - A Boolean formula
  - Pays \$1 if the Boolean formula is satisfied by the final outcome



CPMM: Swap trade for baskets of assets

- Given $\boldsymbol{q} \in \mathbb{R}^n$, some sets $E, E' \subseteq [n]$, and a CPMM $\phi : \mathbb{R}^n \to \mathbb{R}$, we want to support

$$\text{A swap trade for baskets } \boldsymbol{\delta} = 1_E - s \cdot 1_{E'} \in \mathbb{R}^n.$$

- A valid s satisfies

$$\prod_{j \in E'} (q_j - s) = \frac{\prod_{i \in E} q_i \prod_{j \in E'} q_j}{\prod_{i \in E} (q_i + 1)}$$

Designing combinatorial prediction market

- Ω: A large outcome space with $n = |\Omega|$ possible outcomes
- $\mathcal{F} \subseteq 2^{|\Omega|}$: A set system that is a collection of subsets of Ω

- An AMM on $(\Omega, \mathcal{F})$ that can support
  - Price(E): return instantaneous price of any specifies security $E \in \mathcal{F}$;
  - Cost(E, s): return the cost of buying s shares of security on E;
  - Buy(E, s): update the market state after buying s shares of security on E, and return Cost(E, s).

Designing combinatorial prediction market

- Ω: A large outcome space with $n = |\Omega|$ possible outcomes
- $\mathcal{F} \subseteq 2^{|\Omega|}$: A set system that is a collection of subsets of Ω

- An AMM on $(\Omega, \mathcal{F})$ that can support
  - Price(E): return instantaneous price of any specifies security $E \in \mathcal{F}$;
  - Cost(E, s): return the cost of buying s shares of security on E;
  - Buy(E, s): update the market state after buying s shares of security on E, and return Cost(E, s).

- *Can we design efficient algorithms for a prediction market that offers combinatorial security on $(\Omega, \mathcal{F})$ and uses a cost function C?*

Designing combinatorial prediction market

- Ω: A large outcome space with $n = |\Omega|$ possible outcomes
- $\mathcal{F} \subseteq 2^{|\Omega|}$: A set system that is a collection of subsets of Ω

- An AMM on $(\Omega, \mathcal{F})$ that can support
  - Price(E): return instantaneous price of any specifies security $E \in \mathcal{F}$;
  - Cost(E, s): return the cost of buying s shares of security on E;
  - Buy(E, s): update the market state after buying s shares of security on E, and return Cost(E, s).

- *Can we design efficient algorithms for a prediction market that offers combinatorial security on $(\Omega, \mathcal{F})$ and uses a cost function C?*
- AMM for combinatorial markets = Range query range update problem [Hossain et al., 2025]

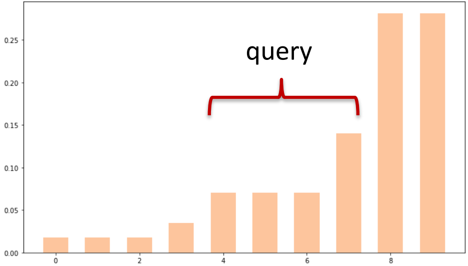
Range query range update (RQRU)

Given $(\Omega, \mathcal{F})$ and initial weights $Q^{(0)} : \Omega \to \mathbb{R}_+$, RQRU performs a sequence of operations for any $E \in \mathcal{F}$ and $S \in \mathbb{R}_+$:

Range query range update (RQRU)

Given $(\Omega, \mathcal{F})$ and initial weights $Q^{(0)} : \Omega \to \mathbb{R}_+$, RQRU performs a sequence of operations for any $E \in \mathcal{F}$ and $S \in \mathbb{R}_+$:

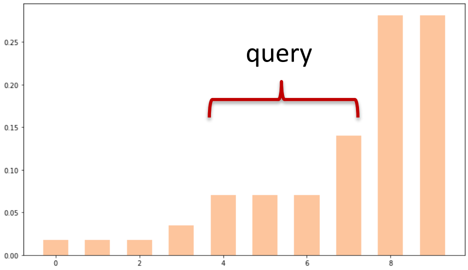
- query(E): return the total weight of a range E, $\sum_{w \in E} Q(w)$;

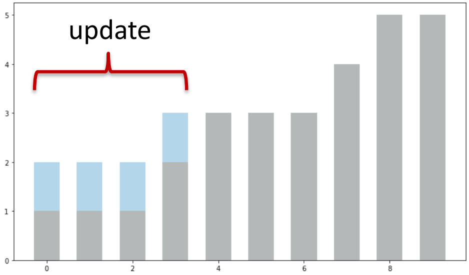


Range query range update (RQRU)

Given $(\Omega, \mathcal{F})$ and initial weights $Q^{(0)} : \Omega \to \mathbb{R}_+$, RQRU performs a sequence of operations for any $E \in \mathcal{F}$ and $S \in \mathbb{R}_+$:

- query(E): return the total weight of a range E, $\sum_{w \in E} Q(w)$;
- update(E, S): update $Q(w) \leftarrow \begin{cases} S \cdot Q(w) & \text{if } w \in E \\ Q(w) & \text{otherwise} \end{cases}$

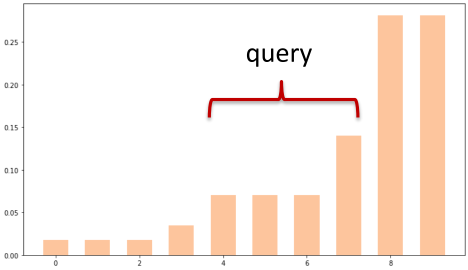


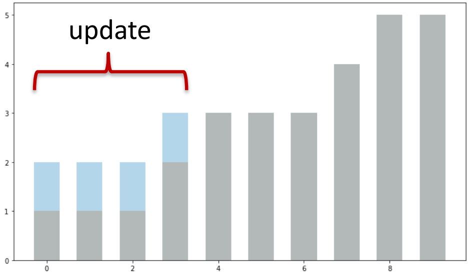


Range query range update (RQRU)

Given $(\Omega, \mathcal{F})$ and initial weights $Q^{(0)} : \Omega \to \mathbb{R}_+$, RQRU performs a sequence of operations for any $E \in \mathcal{F}$ and $S \in \mathbb{R}_+$:

- query(E): return the total weight of a range E, $\sum_{w \in E} Q(w)$;
- update(E, S): update $Q(w) \leftarrow \begin{cases} S \cdot Q(w) & \text{if } w \in E \\ Q(w) & \text{otherwise} \end{cases}$
- We refer to this as $(+, \cdot)$-RQRU





LMSR AMM $\Leftrightarrow$ $(+,\cdot)$-RQRU

Given combinatorial securities in $\mathcal{F}$, a security specifies an event $E \in \mathcal{F}$ and pays \$1 if it happens.

LMSR AMM with $C(\boldsymbol{q}) = \log(\sum_{w\in\Omega} e^{q_w})$ and initial market states $\boldsymbol{q}^{(0)}$ supports [8]

- Price(E): return the price of event E, i.e.,

$$\frac{\sum_{w\in E} e^{q_w}}{\sum_{w\in\Omega} e^{q_w}};$$

- Buy(E, s): update market state $\boldsymbol{q} \leftarrow \boldsymbol{q} + s \cdot 1_E$, and calculate the cost of buying

$$C(\boldsymbol{q} + s \cdot 1_E) - C(\boldsymbol{q}).$$

[8] We assume $b = 1$ for simplicity.

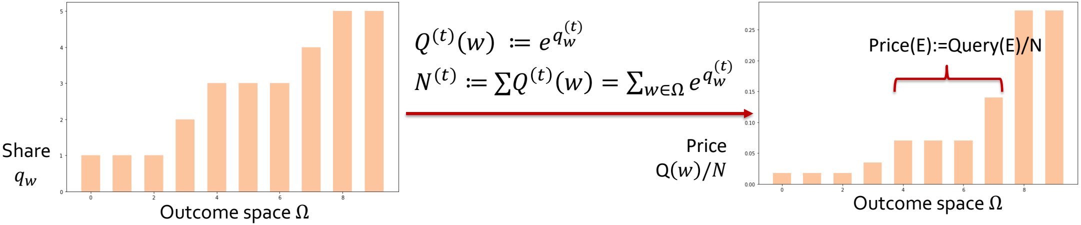
LMSR AMM $\Leftrightarrow$ $(+,\cdot)$-RQRU: Price = Query

Given combinatorial securities in $\mathcal{F}$, a security specifies an event $E \in \mathcal{F}$ and pays \$1 if it happens.

LMSR AMM with $C(\boldsymbol{q}) = \log(\sum_{w\in\Omega} e^{q_w})$ and initial market states $\boldsymbol{q}^{(0)}$ supports

- Price(E): return the price of event E, i.e.,

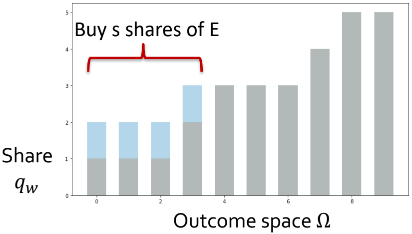
$$\frac{\sum_{w\in E} e^{q_w}}{\sum_{w\in\Omega} e^{q_w}};$$



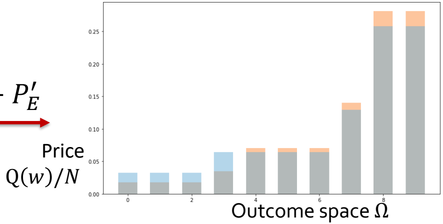
LMSR AMM with $C(\boldsymbol{q}) = \log(\sum_{w\in\Omega} e^{q_w})$ and initial market states $\boldsymbol{q}^{(0)}$ supports

- Buy(E, s): update market state $\boldsymbol{q} \leftarrow \boldsymbol{q} + s \cdot 1_E$, and calculate the cost of buying

$$C(\boldsymbol{q} + s \cdot 1_E) - C(\boldsymbol{q}).$$



1. P_E :=Query(E)
2. Update(E, e^s)
3. P_E' :=Query(E)
4. $N \leftarrow N - P_E + P_E'$



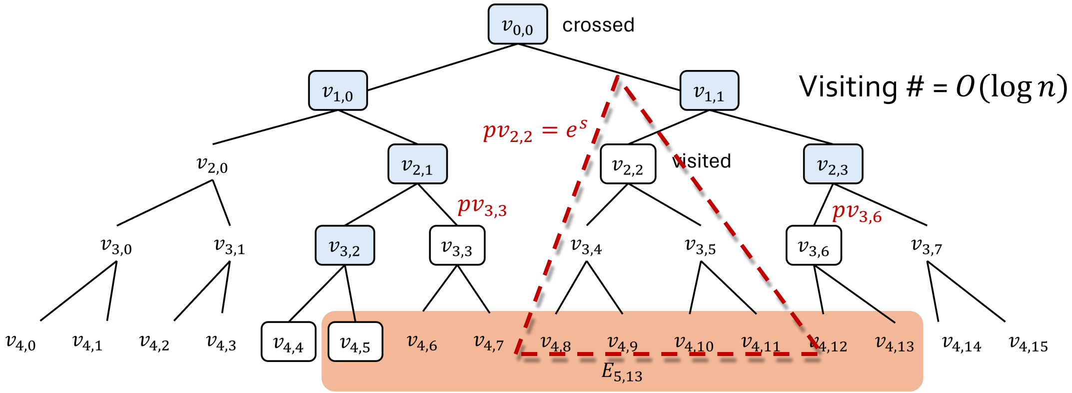
Algorithm and computational complexity: LMSR AMM $\Leftrightarrow$ $(+, \cdot)$-RQRU

Leveraging the connection, we can now use tools from computational geometry to

- Design efficient partition-tree based LMSR on some $\mathcal{F}$ (e.g., interval, d-orthogonal, hyperplane, top L candidates);
- Provide hardness results for some other $\mathcal{F}$ (e.g., pairing, Boolean betting).

Example: A partition tree for interval securities

A partition tree with lazy weight propagation (updating node weights along search path).
Example: Buy 1 share of $[5, 13]$.



Summary: LMSR AMM $\Leftrightarrow$ $(+,\cdot)$-RQRU

If the VC-dimension of $(\Omega, \mathcal{F})$ is infinite, there is no sublinear time algorithm for RQRU using linear space [Chazelle and Welzl, 1989].

Set systems	VC-dim	Run time	Algorithm
Interval	2	$\theta(\log n)$	Interval tree
d-orthogonal set	2d	$O(n^{1-1/d})$	k-d tree
Hyperplane	d+1	$O(n^{1-1/d})$	Partition tree [Chan, 2010]
Permutations	Infinite (increasing in K)	no $o(n)$, $n = K!$	
Boolean	Infinite (increasing in K)	no $o(n)$, $n = 2^K$	

AMM $\Leftrightarrow$ RQRU: Beyond LMSR

Scoring rule	Equivalence	Data structure
Log market scoring rule	$(+, \cdot)$-RQRU	Partition tree
Quadratic market scoring rule	$(+, +)$-RQRU	Partition (segment) tree
γ-power market scoring rule	$(+, \otimes)$-RQRU	Partition tree

CFMM ⇔ RU: Combinatorial swap in DeFi

CFMM	Equivalence	Data structure
Logarithmic	$(+,\cdot)$-RU	Partition tree
Constant sum	$(+,+)$-RU	Partition tree
Geometric mean	$(\cdot,+)$-RU	?

- Logarithmic trading function: $\phi(\boldsymbol{q}) = -\sum_w e^{-q_w/b}$
- Constant sum function: $\phi(\boldsymbol{q}) = \sum_w c_w q_w$
- Geometric mean function: $\phi(\boldsymbol{q}) = \prod_w q_w^{\gamma_w}$

CFMM $\Leftrightarrow$ RU: Combinatorial swap in DeFi

CFMM	Equivalence	Data structure
Logarithmic	$(+,\cdot)$-RU	Partition tree
Constant sum	$(+,+)$-RU	Partition tree
Geometric mean	$(\cdot,+)$-RU	?

- Logarithmic trading function: $\phi(\boldsymbol{q}) = -\sum_w e^{-q_w/b}$
- Constant sum function: $\phi(\boldsymbol{q}) = \sum_w c_w q_w$
- Geometric mean function: $\phi(\boldsymbol{q}) = \prod_w q_w^{\gamma_w}$

Intuition: Decomposable ϕ, i.e., compute $\phi(\boldsymbol{q})$ from q_w and $\phi(\boldsymbol{q}_{-w}, q'_w)$ in constant time. We determine the swap scale through a binary search by querying the trading function ϕ.

Traditional market implementation

- Predetermined discretizations, independent markets

  May suffer
  - Thin market problem
  - Logic inconsistency
  - Arbitrage opportunities

- How about some combinatorial prediction market for large Ω?

  May need to trade off
  - Expressivenss
  - Computational complexity
  - Worst-case loss / liquidity

What price will Bitcoin hit in 2025?

$591,750 Vol. Dec 31, 2025

OUTCOME	% CHANCE		
$1,000,000 $103,025 Vol.	4%	Buy Yes 4.6¢	Buy No 95.8¢
$250,000 $25,090 Vol.	14%	Buy Yes 15¢	Buy No 87¢
$200,000 $53,124 Vol.	23%	Buy Yes 24¢	Buy No 78¢
$150,000 $37,053 Vol.	43%	Buy Yes 44¢	Buy No 59¢
$130,000 $13,674 Vol.	66%	Buy Yes 69¢	Buy No 38¢
$120,000 $11,516 Vol.	74%	Buy Yes 77¢	Buy No 29¢
$110,000 $15,618 Vol.	85%	Buy Yes 86¢	Buy No 17¢

Traditional market implementation

- Predetermined discretizations, independent markets

  May suffer
  - Thin market problem
  - Logic inconsistency
  - Arbitrage opportunities

- How about some combinatorial prediction market for large Ω?

  May need to trade off
  - Expressivenss
  - Computational complexity
  - Worst-case loss / liquidity

What price will Bitcoin hit in 2025?

$591,750 Vol. Dec 31, 2025

OUTCOME	% CHANCE		
$1,000,000 $103,025 Vol.	4%	Buy Yes 4.6¢	Buy No 95.8¢
$250,000 $25,090 Vol.	14%	Buy Yes 15¢	Buy No 87¢
$200,000 $53,124 Vol.	23%	Buy Yes 24¢	Buy No 78¢
$150,000 $37,053 Vol.	43%	Buy Yes 44¢	Buy No 59¢
$130,000 $13,674 Vol.	66%	Buy Yes 69¢	Buy No 38¢
$120,000 $11,516 Vol.	74%	Buy Yes 77¢	Buy No 29¢
$110,000 $15,618 Vol.	85%	Buy Yes 86¢	Buy No 17¢

Challenge: The worst-case loss (e.g., $b \log(n)$) grows with the number of outcomes.

Current Progress

Multi-resolution linearly constrained AMM (LCMM): Interval securities

The tradeoff: the liquidity parameter controls

- How fast the price moves, i.e., $e^{s/b}$;
- The worst-case loss for AMM, e.g., $b \log(n)$.

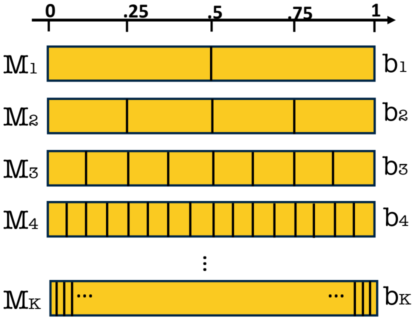
Multi-resolution linearly constrained AMM (LCMM): Interval securities

The tradeoff: the liquidity parameter controls

- How fast the price moves, i.e., $e^{s/b}$;
- The worst-case loss for AMM, e.g., $b\log(n)$.

The intuition
[Dudík et al., 2021, Hossain et al., 2025]

- Use multiple LMSR AMMs with different liquidity parameters to mediate markets offering interval securities at different resolutions (e.g., quarter, week, day, hour markets).
- Achieve constant loss bound by choosing proper liquidity values, e.g., $b_k = O(k^{-2.01})$:
  $\sum_{k=1}^{K} b_k \log(n_k) = \sum_{k=1}^{K} b_k \log(2^k)$

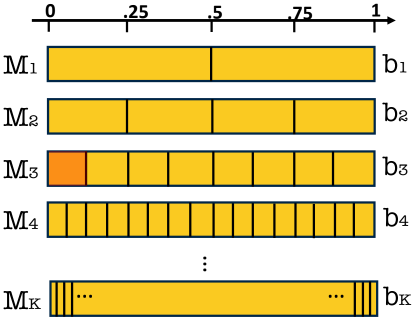


Multi-resolution linearly constrained AMM (LCMM): Interval securities

New challenge: keep prices coherent across different markets.

Buy(E, s)

- Example: Buy$([0, 0.125), 1)$ in M_3.
- Prices become incoherent between M_3 and other markets.



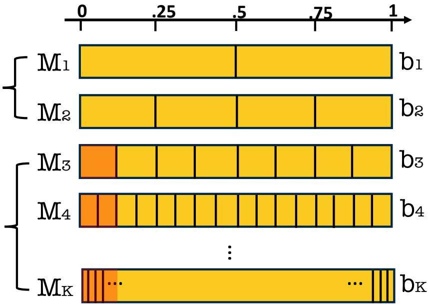
Multi-resolution linearly constrained AMM (LCMM): Interval securities

New challenge: keep prices coherent across different markets.

Buy(E, s)

- Example: Buy($[0, 0.125), 1$) in M_3.
- Prices become incoherent between M_3 and other markets.

- Goal: Remove price incoherence (arbitrage) efficiently across markets.
- Intuition:
  - Split the 1 share among M_3, ..., M_k according to liquidity ratio to maintain price coherence.



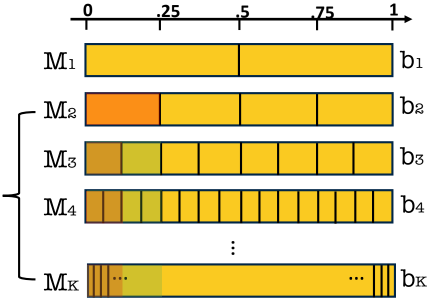
Multi-resolution linearly constrained AMM (LCMM): Interval securities

New challenge: keep prices coherent across different markets.

Buy(E, s)

- Example: Buy($[0, 0.125), 1$) in M_3.
- Prices become incoherent between M_3 and other markets.

- Goal: Remove price incoherence (arbitrage) efficiently across markets.
- Intuition:
  - Split the 1 share among $M_3, ..., M_k$ according to liquidity ratio to maintain price coherence.
  - Remove arbitrage level by level up.



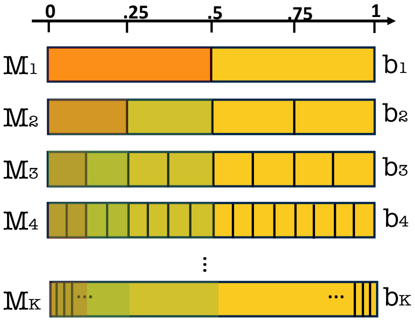
Multi-resolution linearly constrained AMM (LCMM): Interval securities

New challenge: keep prices coherent across different markets.

Buy(E, s)

- Example: Buy($[0, 0.125), 1$) in M_3.
- Prices become incoherent between M_3 and other markets.

- Goal: Remove price incoherence (arbitrage) efficiently across markets.
- Intuition:
  - Split the 1 share among $M_3, ..., M_k$ according to liquidity ratio to maintain price coherence.
  - Remove arbitrage level by level up.



Buy s' share $[0, 0.5)$ in M_1 and split sell s' share among $M_2, ..., M_k$.

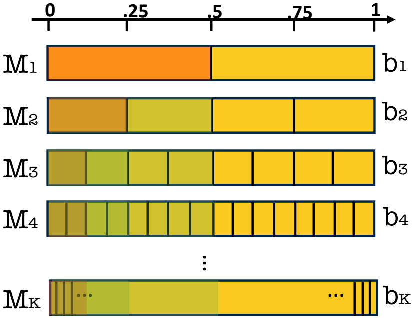
Multi-resolution linearly constrained AMM (LCMM): Interval securities

New challenge: keep prices coherent across different markets.

Multi-resolution LMSR AMM can remove price incoherence (arbitrage) efficiently across markets.

Use a single partition tree and keep track of

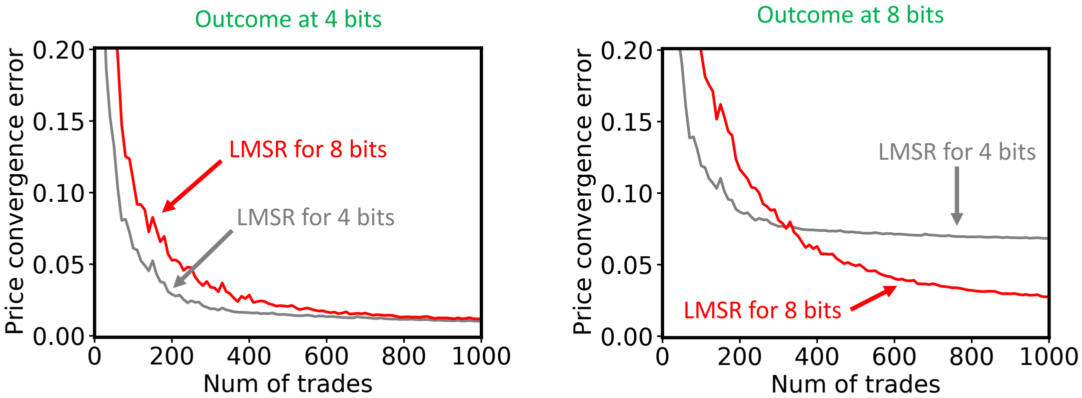
- Trader purchases;
- Automatic purchases made by the AMM for price coherence.



Buy s' share $[0, 0.5)$ in M_1 and split sell s' share among $M_2, \ldots, M_k$.

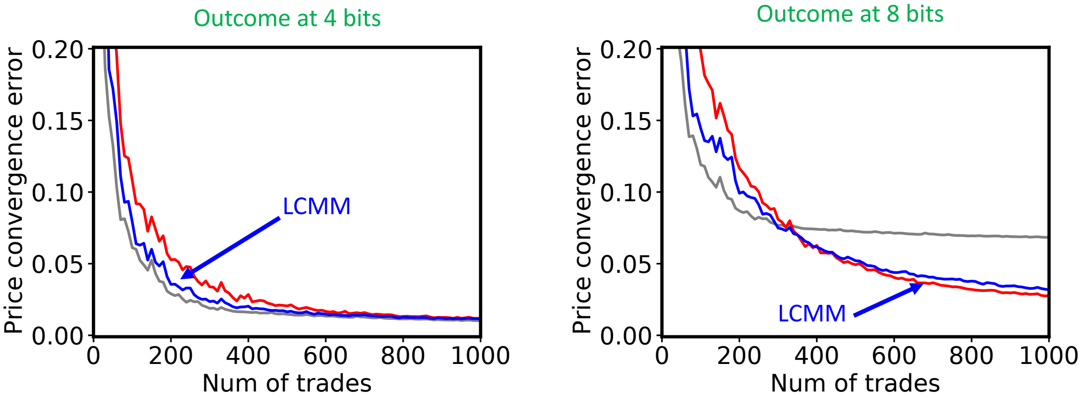
Empirical evaluation: Log-time LMSR vs. Multi-resolution LCMM

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



Empirical evaluation: 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.



Current Progress

Regulatory landscape

- Federal Regulation: Commodity Futures Trading Commission (CFTC)
    - Example: Kalshi, the only fully compliant US exchange.
    - Status: Designated contract market
    - Trades are treated as derivatives/futures (binary options).

Regulatory landscape

- Federal Regulation: Commodity Futures Trading Commission (CFTC)
    - Example: Kalshi, the only fully compliant US exchange.
    - Status: Designated contract market
    - Trades are treated as derivatives/futures (binary options).
- Sweepstakes model
    - Example: Manifold (Sweep)
    - Status: Users buy virtual currency but receive "Sweepcash" as a free bonus. Legally distinct from gambling.

Regulatory landscape

- Federal Regulation: Commodity Futures Trading Commission (CFTC)
  - Example: Kalshi, the only fully compliant US exchange.
  - Status: Designated contract market
  - Trades are treated as derivatives/futures (binary options).
- Sweepstakes model
  - Example: Manifold (Sweep)
  - Status: Users buy virtual currency but receive "Sweepcash" as a free bonus. Legally distinct from gambling.
- DeFi
  - Example: Polymarket
  - Status: Blocked in US; 2022 settlement: Polymarket agreed to block US IP addresses.

Regulatory landscape

- Federal Regulation: Commodity Futures Trading Commission (CFTC)
  - Example: Kalshi, the only fully compliant US exchange.
  - Status: Designated contract market
  - Trades are treated as derivatives/futures (binary options).
- Sweepstakes model
  - Example: Manifold (Sweep)
  - Status: Users buy virtual currency but receive "Sweepcash" as a free bonus. Legally distinct from gambling.
- DeFi
  - Example: Polymarket
  - Status: Blocked in US; 2022 settlement: Polymarket agreed to block US IP addresses.
- Current Legal Conflict
  - *Kalshi v. CFTC (2024):* Current legal battle (Kalshi v. CFTC) regarding whether betting on elections constitutes *gaming* (illegal) or *hedging* (legal).

Open problems and discussions

- Computational complexity of combinatorial markets
  How can we design market mechanisms that allow for expressive betting but remain computationally tractable?

Open problems and discussions

- Computational complexity of combinatorial markets
  How can we design market mechanisms that allow for expressive betting but remain computationally tractable?
- Conditional triviality
  Can we incentivize or elicit accurate forecasting on conditional branches that might never happen (counterfactuals)?

Open problems and discussions

- Computational complexity of combinatorial markets
  How can we design market mechanisms that allow for expressive betting but remain computationally tractable?
- Conditional triviality
  Can we incentivize or elicit accurate forecasting on conditional branches that might never happen (counterfactuals)?
- Manipulation resistance
  Can we design AMMs to differentiate profit-maximizing and outcome-maximizing?

Open problems and discussions

- Computational complexity of combinatorial markets
  How can we design market mechanisms that allow for expressive betting but remain computationally tractable?
- Conditional triviality
  Can we incentivize or elicit accurate forecasting on conditional branches that might never happen (counterfactuals)?
- Manipulation resistance
  Can we design AMMs to differentiate profit-maximizing and outcome-maximizing?
- Privacy-preserving market
  Can we build a market using ZKPs where the AMM can verify the validity/solvency of the trade without knowing who the user is or which outcome they are betting on?

Open problems and discussions

- Computational complexity of combinatorial markets
  How can we design market mechanisms that allow for expressive betting but remain computationally tractable?
- Conditional triviality
  Can we incentivize or elicit accurate forecasting on conditional branches that might never happen (counterfactuals)?
- Manipulation resistance
  Can we design AMMs to differentiate profit-maximizing and outcome-maximizing?
- Privacy-preserving market
  Can we build a market using ZKPs where the AMM can verify the validity/solvency of the trade without knowing who the user is or which outcome they are betting on?
- Capital efficiency & leverage

Buja, A., Stuetzle, W., and Shen, Y. (2005).
Loss functions for binary class probability estimation and classification: Structure and applications.

Chan, T. M. (2010).
Optimal partition trees.
In *Proceedings of the twenty-sixth annual symposium on Computational geometry*, pages 1–10.

Chazelle, B. and Welzl, E. (1989).
Quasi-optimal range searching in spaces of finite vc-dimension.
Discrete & Computational Geometry, 4.

Chen, Y., Fortnow, L., Lambert, N. S., Pennock, D. M., and Wortman, J. (2008).
Complexity of combinatorial market makers.
CoRR, abs/0802.1362.

Chen, Y., Fortnow, L., Nikolova, E., and Pennock, D. M. (2007).
Combinatorial betting.

SIGecom Exch., 7(1):61–64.

Chen, Y. and Pennock, D. M. (2012).
A utility framework for bounded-loss market makers.
arXiv preprint arXiv:1206.5252.

Chen, Y. and Yu, F. (2021).
Optimal scoring rule design.
CoRR, abs/2107.07420.

Dudík, M., Wang, X., Pennock, D. M., and Rothschild, D. M. (2021).
Log-time prediction markets for interval securities.
CoRR, abs/2102.07308.

Frongillo, R. and Kash, I. (2014).
General truthfulness characterizations via convex analysis.
In *International Conference on Web and Internet Economics*, pages 354–370. Springer.

Frongillo, R. and Witkowski, J. (2017).
A geometric perspective on minimal peer prediction.

ACM Trans. Econ. Comput., 5(3).

Frongillo, R. M., Papireddygari, M., and Waggoner, B. (2024).
An axiomatic characterization of cfmms and equivalence to prediction markets.
In Guruswami, V., editor, *15th Innovations in Theoretical Computer Science Conference, ITCS 2024, January 30 to February 2, 2024, Berkeley, CA, USA*, volume 287 of *LIPIcs*, pages 51:1–51:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Gneiting, T. and Raftery, A. E. (2007).
Strictly proper scoring rules, prediction, and estimation.
Journal of the American Statistical Association, 102(477):359–378.

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

Hanson, R. D. (2007).
Logarithmic market scoring rules for modular combinatorial information aggregation.
Journal of Prediction Markets, 1(1):1–15.

Hartline, J. D., Li, Y., Shan, L., and Wu, Y. (2020).
Optimization of scoring rules.
CoRR, abs/2007.02905.

Hossain, P. S., Wang, X., and Yu, F. (2025).
Designing automated market makers for combinatorial securities: A geometric viewpoint.
In Azar, Y. and Panigrahi, D., editors, *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2025, New Orleans, LA, USA, January 12-15, 2025*, pages 1329–1365. SIAM.

Kleinberg, B., Leme, R. P., Schneider, J., and Teng, Y. (2023).
U-calibration: Forecasting for an unknown agent.
In *The Thirty Sixth Annual Conference on Learning Theory*, pages 5143–5145. PMLR.

Kong, Y. and Schoenebeck, G. (2019).
An information theoretic framework for designing information elicitation mechanisms that reward truth-telling.
ACM Trans. Econ. Comput., 7(1).

Lambert, N. and Shoham, Y. (2009).
Eliciting truthful answers to multiple-choice questions.
In *Proceedings of the 10th ACM Conference on Electronic Commerce*, EC '09, page 109–118, New York, NY, USA. Association for Computing Machinery.

Lambert, N. S., Langford, J., Wortman Vaughan, J., Chen, Y., Reeves, D. M., Shoham, Y., and Pennock, D. M. (2015).
An axiomatic characterization of wagering mechanisms.
Journal of Economic Theory, 156:389–416.
Computer Science and Economic Theory.

Liu, Y., Wang, J., and Chen, Y. (2023).
Surrogate scoring rules.
ACM Transactions on Economics and Computation, 10(3):1–36.

Lu, Y., Xu, S., Zhang, Y., Kong, Y., and Schoenebeck, G. (2024).
Eliciting informative text evaluations with large language models.

Miller, N., Resnick, P., and Zeckhauser, R. (2005).

Eliciting informative feedback: The peer-prediction method.
Management Science, pages 1359–1373.

Prelec, D. (2004).
A Bayesian Truth Serum for subjective data.
Science, 306(5695):462–466.

Rochet, J. C. (1985).
The taxation principle and multi-time hamilton-jacobi equations.
Journal of Mathematical Economics, 14(2):113–128.

Schoenebeck, G. and Yu, F. (2020).
Learning and strongly truthful multi-task peer prediction: A variational approach.
CoRR, abs/2009.14730.

Schoenebeck, G. and Yu, F.-Y. (2023).
Two strongly truthful mechanisms for three heterogeneous agents answering one question.
ACM Transactions on Economics and Computation, 10(4):1–26.

Wang, X., Pennock, D. M., Devanur, N. R., Rothschild, D. M., Tao, B., and Wellman, M. P. (2021).
Designing a combinatorial financial options market.
In *Proceedings of the 22nd ACM Conference on Economics and Computation (EC)*, page 864–883.

Xia, Z. (2025).
Expert incentives under partially contractible states.