Hanson’s Market Scoring Rules

Robin Hanson, **Logarithmic Market Scoring Rules for Modular Combinatorial Information Aggregation**, 2002.

Robin Hanson, **Combinatorial Information Market Design**, 2003.

Proper Scoring Rules

- Report a probability estimate $\mathbf{r}$, get payment $s_i(\mathbf{r})$ if outcome i happens.
- Risk-neutral agents report their beliefs accurately as this maximizes expected payoff (example: $s(\mathbf{r}) = a + b \log(r_i)$).
- Problem:
  - Pooling opinions is difficult

Continuous Double Auction Information Markets

- Like scoring rules, give people incentives to be honest.
- Produces common estimates that combines all information through repeated interaction among rational agents.
- Problems:
  - Irrational to participate
  - Thin markets

Hanson’s Market Scoring Rule (MSR)

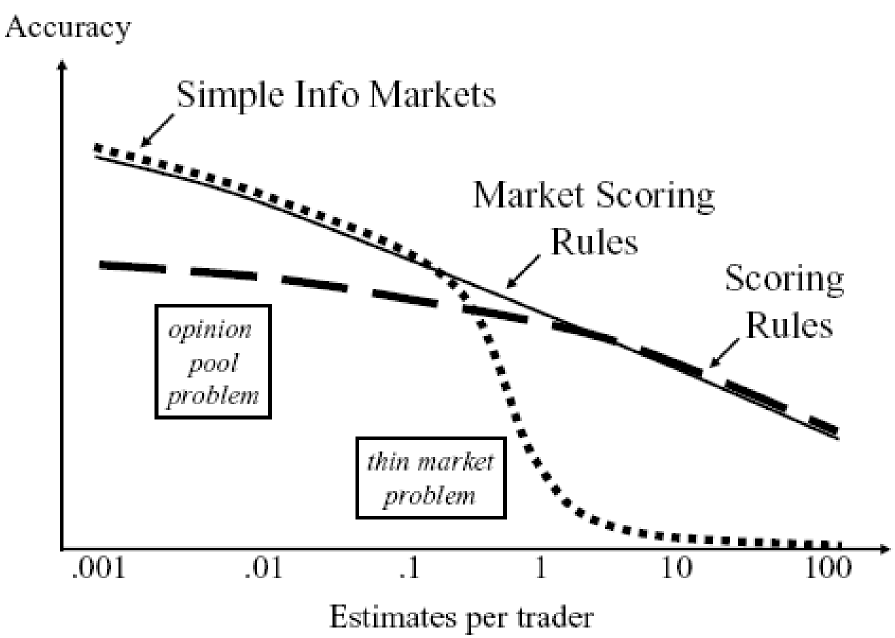
- Market maker establishes initial distribution. Any trader can report a new distribution.
- In making the new report, the agent will be responsible for the scoring rule payment according to the last report.
- Agent receives scoring rule payment according to his new report and maximizes expected utility by reporting honestly.
- Market maker is responsible only for paying difference between his initial report $\mathbf{r_0}$ and the final report $\mathbf{r_T}$.
- Formally:

$$x_i = \Delta s_i(\vec{r}, \vec{\rho}) = s_i(\vec{r}) - s_i(\vec{\rho})$$

where x_i is the agent’s reward, s_i is some proper scoring rule, **r** is the agent’s report, and ❒ is the current probability distribution

Why use a MSR?

- Subsidized market makes it rational to participate
- Increased liquidity even with thin markets
- Ability to express more outcomes without requiring matched traders



Logarithmic Market Scoring Rule (LMSR)

- Proper scoring rule

$$s_i = \quad a_i + b\log(r_i)$$

- b measures liquidity, potential loss of market maker – larger b means traders can buy more shares at or near the current price without causing massive price swings
- Principal’s expected cost given initial report $\mathbf{r_0}$ = ($\blacksquare_1$, $\blacksquare_2$, … $\blacksquare_n$) is the entropy of the initial distribution

$$-b\sum_i \pi_i \log(\pi_i)$$

- We can reformulate the LMSR in terms of "buying" and "selling" shares instead of changing the probability distribution
- Inkling.com implements this type of automated market maker

Changing the Distribution = Buying/Selling Shares

$$x_i = \Delta s_i(\vec{r}, \vec{\rho}) = s_i(\vec{r}) - s_i(\vec{\rho})$$

A sequence of reports from $\vec{r}_0$ to $\vec{r}_T$ can be split at no additional cost into smaller movements from $\vec{r}_{t-1}$ to $\vec{r}_t$ (movements $d\vec{r}$ along a line $\vec{r}(t)$ as t varies continuously)

Let $q_i = \frac{dr_i}{dt}$ be the rate at which the agent changes his report.

Let $y_i = \frac{dx_i}{dt} = \sum_j q_j \nabla_j s_i$ be the rate of change in his asset amounts.

For an agent with beliefs **p**, the rate of change in his expected payoff is:

$$\frac{d}{dt}\sum_i p_i x_i = \sum_i p_i y_i = \sum_i p_i \sum_j q_j \nabla_j s_i(\vec{r}) = \sum_j q_j (\sum_i p_i \nabla_j s_i(\vec{r}))$$

For **r** = **p**, this has zero expected value (notice FOC for proper scoring rule). Thus, assets exchanged as an agent changes one's report are locally fair at current "market" prices **r**.

Changing the Distribution = Buying/Selling Shares

- So, we can think of a market scoring rule as a automated inventory-based market maker with:
  - Zero bid-ask spread for infinitesimal trades (which we showed in the previous slide)
  - An internal state described by inventory of assets $\vec{x}$
  - Instantaneous price:

$$\vec{p} = \vec{m}(\vec{x}), \text{ where } \sum_i m_i(\vec{x}) = 1$$

  - Market maker will accept any fair bet $d\vec{x} = \vec{y}\, dt$ s.t.

$$\sum_i y_i\, m_i(\vec{x}) = 0$$

    and any integral of infinitesimal trades.

Example: LMSR Cost Function

- Consider a two-outcome space $\mathbf{q} = (q_1,q_2)$ and a proper scoring rule $s_i(\mathbf{p}) = b \log(p_i)$
- Instantaneous price of q1: $$\frac{e^{\frac{q_1}{b}}}{e^{\frac{q_1}{b}} + e^{\frac{q_2}{b}}}$$
- Cost function: $C(q_1, q_2) = b \log(e^{\frac{q_1}{b}} + e^{\frac{q_2}{b}})$
- Market maker keeps track of shares outstanding to quote prices.
- If I want to buy 15 shares of q1, and there are 10 shares each of q1, q2 outstanding, this would cost: C(25,10) – C(10,10)

Modularity

- How well do MSR preserve conditional independence relations?
- Example: placing a bet on conditional event A given B should not change P(B) or P(C) for some event C unrelated to how A might depend on B
- Logarithmic rule bets on A given B preserve P(B), and for any event C, preserve $P(C|AB)$, $P(C|A^cB)$, and $P(C|B^c)$
- Turns out LMSR is uniquely able to do this

Combinatorial Product Space

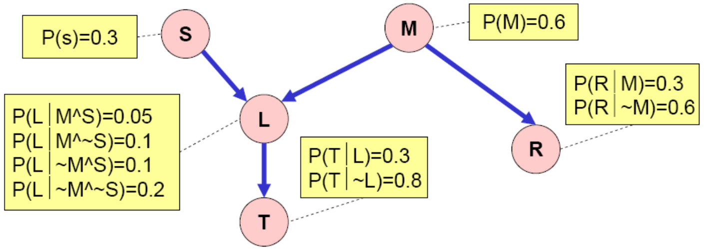
- Given N variables each with V outcomes, a single market scoring rule can make trades on any of the V^N possible states, or any of the $2^{(V^N)}$ possible events.
- Creating a data structure to explicitly store the probability of every such state is unfeasible for large values of N.
- Computational complexity of updating prices and assets is worse than polynomial in the worst case (NP-complete).

Ways to Deal with Large State Space

- Limit probability distribution
  - Example: Bayes Net – variables organized by a directed graph where each variable has a set of parents. Probability of a state i can be written as:

$$p_i = \prod_{\alpha} p(v_\alpha(i) \mid \{v_k(i)\}_{k \in \mathcal{P}_\alpha})$$

    which states that value of a variable in a state i can be computed based on the conditional dependencies with all parents.
  - For a sparse network, this makes it easier to store the data as we need to keep track of fewer variables



Ways to Deal with Large State Space

  - Problem: Supporting bets on conditional probabilities not specified in net or unconditional probabilities – harder to do unless you have “nearly” singly connected Bayes Net
  - Using an approximation algorithm to calculate probabilities in a more complicated Bayes Nets runs risk of opening new arbitrage opportunities
- Use Multiple Market Makers
  - Example: Combine MSR that represents probabilities via a general sparse Bayes net and a MSR that deals only with the unconditional probabilities
  - Problem: Arbitrage opportunities across patrons, but the amount of loss is now bounded (since we can bound the loss for each rule).

Open Questions

- What's the most effective way to set b, the liquidity constraint?
  - High b desirable for thin market, low b desirable for thick market.
- How can we deal with large state space of allowing combinatorial outcomes?
- Does LMSR work as well as traditional prediction markets empirically?
- Do there exist circumstances where it makes strategic sense to bluff or hide information?