Announcements

- HW 3 is out, due Nov 5'th
- Project instruction is out
  - Format: proposal (5') + presentation (10') + report (25')
  - Proposal due Nov 7'th -- mainly to check you formed a team and have some ideas about what to do
  - We have some suggested topics, but you are more encouraged to find your own

CS6501: Topics in Learning and Game Theory (Fall 2019)

Scoring Rules

Instructor: Haifeng Xu

Outline

- Recap: Prediction Markets
- Scoring Rule and its Characterization
- Connection to Prediction Markets

Prediction Markets

A prediction market is a financial market that is designed for event prediction via information aggregation

➢ Payoffs of the traded contract are determined by outcomes of future events

$1 iff e_1 · · · $1 iff e_n

contracts

We design a market maker by specifying the payment for bundles of contracts.

Example: Logarithmic Market Scoring Rule (LMSR [Hanson 03, 06])

$1 iff e_1	. . .	$1 iff e_n

- Define value function ($q = (q_1, \cdots, q_n)$ is current sales quantity)

$$V(q) = b \log \sum_{j\in[n]} e^{q_j/b}$$

Parameter b adjusts liquidity

- Price function

$$p_i(q) = \frac{e^{q_i/b}}{\sum_{j\in[n]} e^{q_j/b}} = \frac{\partial V(q)}{\partial q_i}$$

- To buy $x \in \mathbb{R}^n$ amount, a buyer pays: $V(q+x) - V(q)$
  - Negative x_i's mean selling contracts to MM
  - Negative payment means market maker pays the buyer
  - Market starts with $V(0) = b \log n$

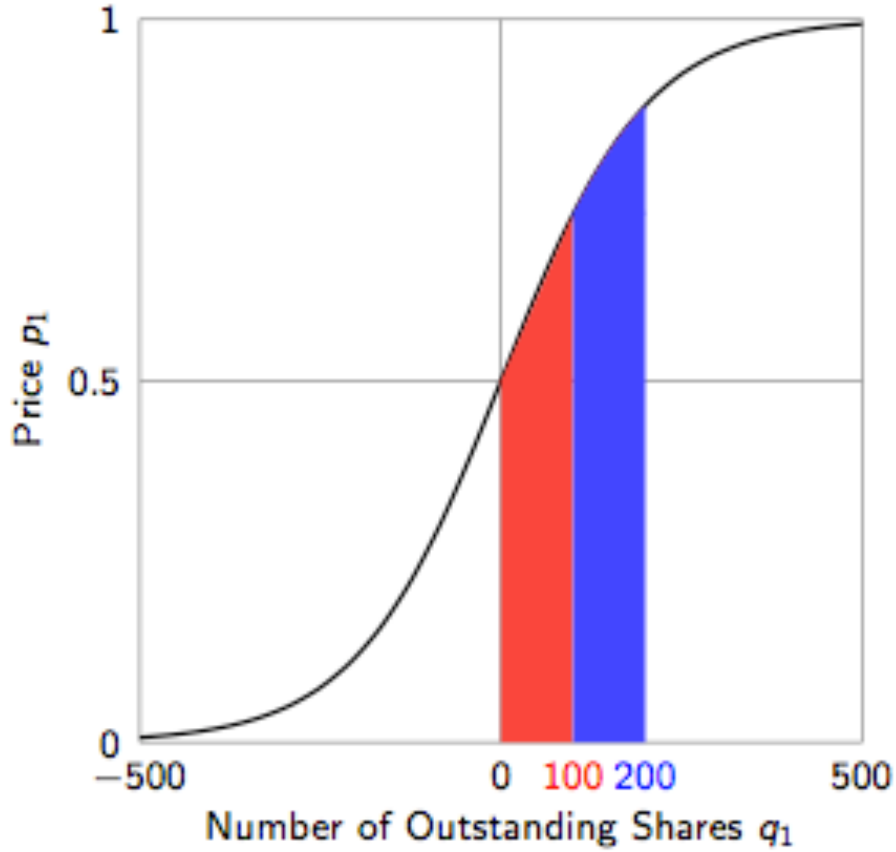
Properties of LMSR

Fact. The optimal amount an expert purchases is the amount that moves the market price to her belief λ. Her expected utility of purchasing this amount is always non-negative.

- I.e., should purchase x^* such that $\frac{\partial V(q+x^*)}{\partial x_i^*} = \lambda_i$
- Market efficiency

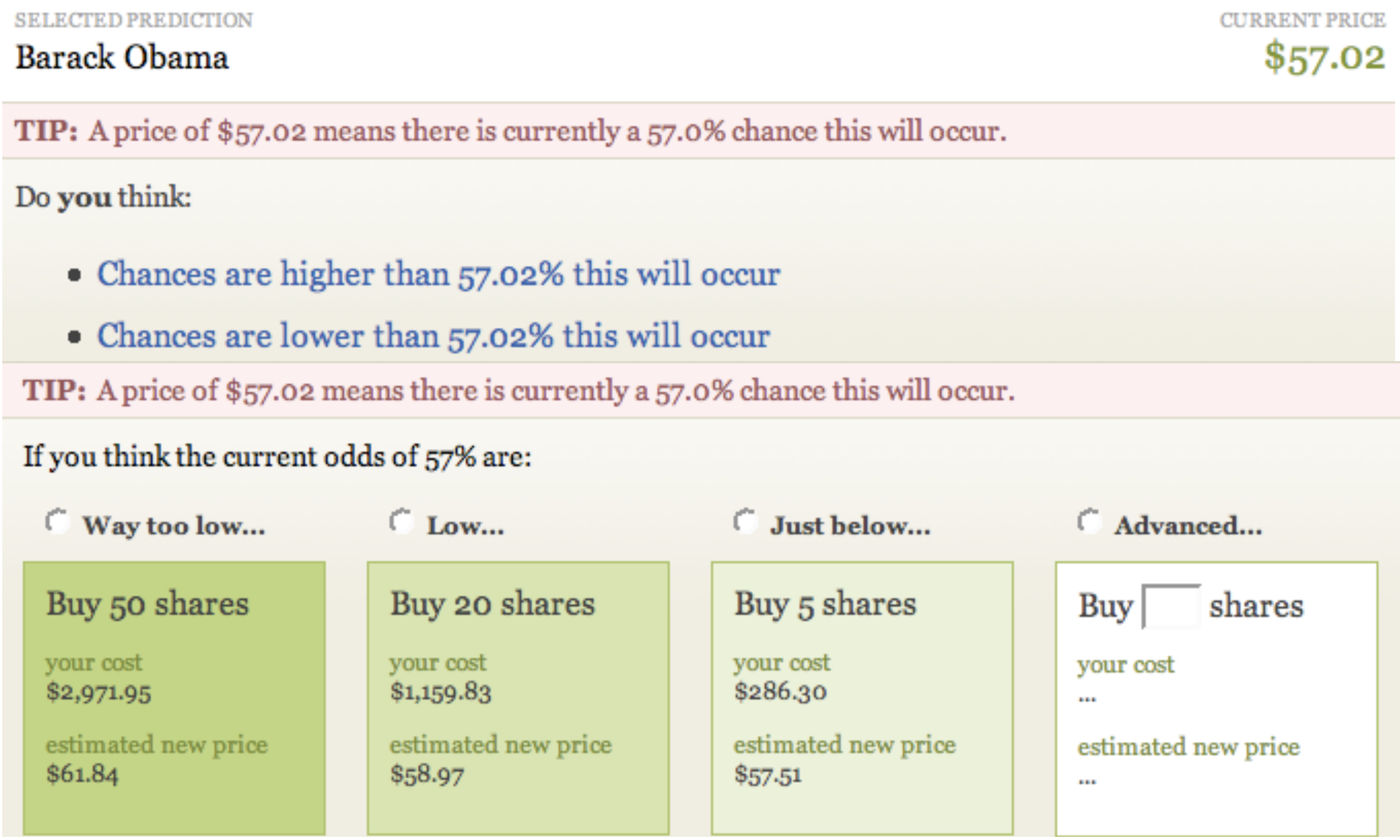
Fact. Worst case market maker loses is $b \log n$ (i.e., bounded).

Price Curve as a Function of Share Quantities



Examples of LMSR in Practice

- Has been implemented by several prediction markets
  - E.g., InklingMarkets, Washington Stock Exchange, BizPredict, Net Exchange, and (reportedly) at YooNew.



Markets can potentially be a very effective forecasting tool

Big on-going project: “replication market” for DARPA SCORE project

DARPA DEFENSE ADVANCED RESEARCH PROJECTS AGENCY

EXPLORE BY TAG

ABOUT US / OUR RESEARCH / NEWS / EVENTS / WORK WITH US /

Defense Advanced Research Projects Agency › Program Information

Systematizing Confidence in Open Research and Evidence (SCORE)

Dr. Adam Russell

The Department of Defense (DoD) often leverages social and behavioral science (SBS) research to design plans, guide investments, assess outcomes, and build models of human social systems and behaviors as they relate to national security challenges in the human domain. However, a number of recent empirical studies and meta-analyses have revealed that many SBS results vary dramatically in terms of their ability to be independently reproduced or replicated, which could have real-world implications for DoD’s plans, decisions, and models. To help address this situation, DARPA’s Systematizing Confidence in Open Research and Evidence (SCORE) program aims to develop and deploy automated tools to assign "confidence scores" to different SBS research results and claims. Confidence scores are quantitative measures that should enable a DoD consumer of SBS research to understand the degree to which a particular claim or result is likely to be reproducible or replicable. These tools will assign explainable confidence scores with a reliability that is equal to, or better than, the best current human expert methods. If successful, SCORE will enable DoD personnel to quickly calibrate the level of confidence they should have in the reproducibility and replicability of a given SBS result or claim, and thereby increase the effective use of SBS literature and research to address important human domain challenges, such as enhancing deterrence, enabling stability, and reducing extremism.

Connection between LMSR and Exponential Weight Updates (EWU)

Recap: Exponential Weight Update

- Played for T rounds; each round selects an action $i \in [n]$
- Maintains weights over n actions: $w_t(1), \cdots, w_t(n)$
- Observe cost vector c_t, and update $w_{t+1}(i) = w_t(i) \cdot e^{-\epsilon c_t(i)}, \forall i \in [n]$

Action 1, $w_t(1)$

Action 2, $w_t(2)$

. . .

Action n, $w_t(n)$

Recap: Exponential Weight Update

- Played for T rounds; each round selects an action $i \in [n]$
- Maintains weights over n actions: $w_t(1), \cdots, w_t(n)$
- Observe cost vector c_t, and update $w_{t+1}(i) = w_t(i) \cdot e^{-\epsilon c_t(i)}, \forall i \in [n]$

Action 1, $w_t(1)$

Action 2, $w_t(2)$

. . .

Action n, $w_t(n)$

$$w_{t+1}(i) = w_t(i) \cdot e^{-\epsilon c_t(i)}$$

$$= [w_{t-1}(i) \cdot e^{-\epsilon c_{t-1}(i)}] \cdot e^{-\epsilon c_t(i)}$$

$$= \cdots = e^{-\epsilon C_t(i)} \text{ where } C_t(i) = \sum_{\tau \le t} c_\tau(i)$$

Recap: Exponential Weight Update

- Played for T rounds; each round selects an action $i \in [n]$
- Maintains weights over n actions: $w_t(1), \cdots, w_t(n)$
- Observe cost vector c_t, and update $w_{t+1}(i) = w_t(i) \cdot e^{-\epsilon c_t(i)}, \forall i \in [n]$
- At round $t + 1$, select action i with probability

$$\frac{w_t(i)}{W_t} = \frac{e^{-\epsilon C_t(i)}}{\sum_{j \in [n]} e^{-\epsilon C_t(j)}}$$

where $C_t = \sum_{\tau \le t} c_t$ is the accumulated cost vector

Recap: Exponential Weight Update

- Played for T rounds; each round selects an action $i \in [n]$
- Maintains weights over n actions: $w_t(1), \cdots, w_t(n)$
- Observe cost vector c_t, and update $w_{t+1}(i) = w_t(i) \cdot e^{-\epsilon c_t(i)}, \forall i \in [n]$
- At round $t+1$, select action i with probability

$$\frac{w_t(i)}{W_t} = \frac{e^{-\epsilon C_t(i)}}{\sum_{j \in [n]} e^{-\epsilon C_t(j)}}$$

where $C_t = \sum_{\tau \le t} c_t$ is the accumulated cost vector

This looks very much like the price function in LMSR (q is the accumulated sales quantity)

$$p_i = \frac{e^{q_i/b}}{\sum_{j \in [n]} e^{q_j/b}}$$

EWU vs LMSR

- Exponential Weight Update
  - n actions
  - Maintain weight $w_t(i)$
  - Total cost $C_T(i) = \sum_{t \le T} c_t(i)$
  - Select i with prob

$$p_i = \frac{e^{-\epsilon C_t(i)}}{\sum_{j \in [n]} e^{-\epsilon C_t(j)}}$$

  - Weights reflect how good an action is
  - Care about worst case regret

$$C_T(\mathrm{Alg}) - \min_i C_T(i)$$

- LMSR
  - n contracts (i.e., outcomes)
  - Maintain prices $p(i)$
  - Total shares sold $q(i)$
  - Price of contract i

$$p_i = \frac{e^{q_i/b}}{\sum_{j \in [n]} e^{q_j/b}}$$

  - Prices reflect how probable is an event
  - Care about worst case MM loss

$$(\text{\$ received}) - \max_i q(i)$$

- LMSR is just one particular automatic MM
  - Similar relation holds for other market markers and no-regret learning algorithms (see [Chen and Vaughan 2010])

- Next: will study other "good" scoring rules, and see why they work

Outline

- Recap: Prediction Markets
- Scoring Rule and its Characterization
- Connection to Prediction Markets

Consider a Simpler Setting

- We (designer) want to learn the distribution of random var $E \in [n]$
  - E will be sampled in the future
- We have no samples from E; Instead, we have an expert/predictor who has a predicted distribution $\lambda \in \Delta_n$
- We want to incentivize the expert to truthfully report λ

Consider a Simpler Setting

- We (designer) want to learn the distribution of random var $E \in [n]$
  - E will be sampled in the future
- We have no samples from E; Instead, we have an expert/predictor who has a predicted distribution $\lambda \in \Delta_n$
- We want to incentivize the expert to truthfully report λ

Example
- E is whether UVA will win NCAA title in 2020
- Expert is the UVA coach

- Expert's prediction does not need to be perfect
  - But, better than the designer who knows nothing
- Assume expert will not give you truthful info for free

Idea: "Score" Expert's Report

Will reward the expert certain amount $S(i; p)$ where:
(1) p is the expert's report (does not have to equal λ);
(2) $i \in [n]$ is the event realization

Not like a prediction market yet, but will see later they are related

Idea: "Score" Expert's Report

Will reward the expert certain amount $S(i; p)$ where:
(1) p is the expert's report (does not have to equal λ);
(2) $i \in [n]$ is the event realization

Q: what is the expert's expected utility?

- Expert believes $i \sim \lambda$
- Expected utility $\mathbb{E}_{i \sim \lambda} S(i; p) = \sum_{i \in [n]} \lambda_i \cdot S(i; p) = S(\lambda; p)$

Idea: "Score" Expert's Report

Will reward the expert certain amount $S(i; p)$ where:
(1) p is the expert's report (does not have to equal λ);
(2) $i \in [n]$ is the event realization

Q: what is the expert's expected utility?

- Expert believes $i \sim \lambda$
- Expected utility $\mathbb{E}_{i\sim\lambda} S(i; p) = \sum_{i\in[n]} \lambda_i \cdot S(i; p) = S(\lambda; p)$

Q: what $S(i; p)$ function can elicit truthful report λ?

- When expert finds that $\lambda = \arg\max_{p\in\Delta_n} [\sum_{i\in[n]} \lambda_i \cdot S(i; p)]$
- Ideally, λ is the unique maximizer

Proper Scoring Rules

Definition. The "scoring rule" $S(i;p)$is [strictly] proper if truthful report $p = \lambda$ [uniquely] maximizes expected utility $S(\lambda;p)$.

- Expert is incentivized to report truthfully iff $S(e;p)$ is proper

Observations.

1. $S(i;p) = 0$ is a trivial proper scoring fnc
2. Proper scores are closed under affine transformation
   - I.e., if $S(i;p)$ is [strictly] proper, so is $\alpha \cdot S(i;p) + \beta$ for any constant $\alpha \neq 0, \beta$

- Thus, typically, strict properness is desired

Examples of Scoring Rules

Example 1 [Log Scoring Rule]
- $S(i; p) = \log p_i$
- $S(\lambda; p) = \sum_{i \in [n]} \lambda_i \cdot \log p_i$

- Negative, but okay – can always add a constant
- Properness requires $\lambda = \arg\max_{p \in \Delta_n} S(\lambda; p)$

$$S(\lambda; p) = \sum_{i \in [n]} \lambda_i \cdot \log p_i$$

$$= \sum_{i \in [n]} \lambda_i [\log p_i - \log \lambda_i] + \sum_{i \in [n]} \lambda_i \log \lambda_i$$

Examples of Scoring Rules

Example 1 [Log Scoring Rule]

- $S(i; p) = \log p_i$
- $S(\lambda; p) = \sum_{i \in [n]} \lambda_i \cdot \log p_i$

- Negative, but okay – can always add a constant
- Properness requires $\lambda = \arg\max_{p \in \Delta_n} S(\lambda; p)$

$$S(\lambda; p) = \sum_{i \in [n]} \lambda_i \cdot \log p_i$$

$$= \sum_{i \in [n]} \lambda_i [\log p_i - \log \lambda_i] + \sum_{i \in [n]} \lambda_i \log \lambda_i$$

$$= -\sum_{i \in [n]} \lambda_i \cdot \log \frac{\lambda_i}{p_i} - Entrop(\lambda)$$

Examples of Scoring Rules

Example 1 [Log Scoring Rule]

- $S(i; p) = \log p_i$
- $S(\lambda; p) = \sum_{i \in [n]} \lambda_i \cdot \log p_i$

- Negative, but okay – can always add a constant
- Properness requires $\lambda = \arg\max_{p \in \Delta_n} S(\lambda; p)$

$$
\begin{aligned}
S(\lambda; p) &= \sum_{i \in [n]} \lambda_i \cdot \log p_i \\
&= \sum_{i \in [n]} \lambda_i [\log p_i - \log \lambda_i] + \sum_{i \in [n]} \lambda_i \log \lambda_i \\
&= - \underbrace{\sum_{i \in [n]} \lambda_i \cdot \log \frac{\lambda_i}{p_i}} - Entrop(\lambda)
\end{aligned}
$$

KL-divergence $KL(\lambda; p)$ (a.k.a. relative entropy)

- Measures the distance between two distributions
- Always non-negative, and equals 0 only when $p = \lambda$

Examples of Scoring Rules

Example 1 [Log Scoring Rule]

- $S(i;p) = \log p_i$
- $S(\lambda;p) = \sum_{i\in[n]} \lambda_i \cdot \log p_i$

- Negative, but okay – can always add a constant
- Properness requires $\lambda = \arg\max_{p\in\Delta_n} S(\lambda;p)$

$$S(\lambda;p) = \sum_{i\in[n]} \lambda_i \cdot \log p_i$$

$$= \sum_{i\in[n]} \lambda_i [\log p_i - \log \lambda_i] + \sum_{i\in[n]} \lambda_i \log \lambda_i$$

$$= -\sum_{i\in[n]} \lambda_i \cdot \log \frac{\lambda_i}{p_i} - Entrop(\lambda)$$

- p should minimize distance $KL(\lambda;p)$, which is achieved at $p = \lambda$
- Log scoring rule is strictly proper

Examples of Scoring Rules

Example 2 [Quadratic Scoring Rule]

- $S(i;p) = 2p_i - \sum_{j\in[n]} p_j^2$
- $S(\lambda;p) = \sum_{i\in[n]} \lambda_i [2p_i - \sum_{j\in[n]} p_j^2]$

Examples of Scoring Rules

Example 2 [Quadratic Scoring Rule]

- $S(i; p) = 2p_i - \sum_{j\in[n]} p_j^2$
- $S(\lambda; p) = \sum_{i\in[n]} \lambda_i [2p_i - \sum_{j\in[n]} p_j^2]$

$$
\begin{aligned}
S(\lambda; p) &= \sum_{i\in[n]} \lambda_i [2p_i - \sum_{j\in[n]} p_j^2] \\
&= \sum_{i\in[n]} 2\lambda_i p_i - \left(\sum_{i\in[n]} \lambda_i\right) \cdot \sum_{j\in[n]} p_j^2 \\
&= \sum_{i\in[n]} 2\lambda_i p_i - \sum_{i\in[n]} p_i^2 \\
&= -\sum_{i\in[n]} [p_i - \lambda_i]^2 + \sum_{i\in[n]} \lambda_i^2
\end{aligned}
$$

- Prediction p should minimize l_2-distance between p and λ
- $p_i = \lambda_i$ is the unique maximizer of $S(\lambda; p)$
- Quadratic scoring rule is also strictly proper

Examples of Scoring Rules

Example 3 [Linear Scoring Rule]

- $S(i; p) = p_i$
- $S(\lambda; p) = \sum_{i \in [n]} \lambda_i p_i$

- Linear scoring rule turns out to be not proper (verify it after class)

What $S(i; p)$ Are Proper?

Theorem. The scoring rule $S(i; p)$ is (strictly) proper if and only if there exists a (strictly) convex function $G: \Delta_n \to \mathbb{R}$ such that

$$S(i; p) = G(p) + \nabla G(p)(e_i - p)$$

basis vector

What $S(i;p)$ Are Proper?

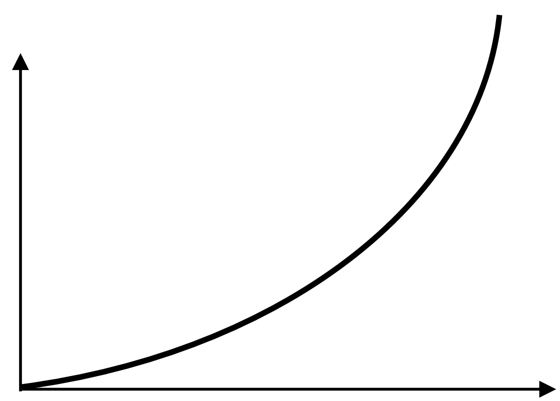
Theorem. The scoring rule $S(i;p)$ is (strictly) proper if and only if there exists a (strictly) convex function $G: \Delta_n \rightarrow \mathbb{R}$ such that

$$S(i;p) = G(p) + \nabla G(p)(e_i - p)$$

basis vector

Recall $G(p)$ is convex if for any $\alpha \in [0,1]$

$$\alpha G(p) + (1-\alpha)G(q) \geq G(\ \alpha p + (1-\alpha)q\ )$$



What $S(i;p)$ Are Proper?

Theorem. The scoring rule $S(i;p)$ is (strictly) proper if and only if there exists a (strictly) convex function $G:\Delta_n \to \mathbb{R}$ such that

$$S(i;p) = G(p) + \nabla G(p)(e_i - p)$$

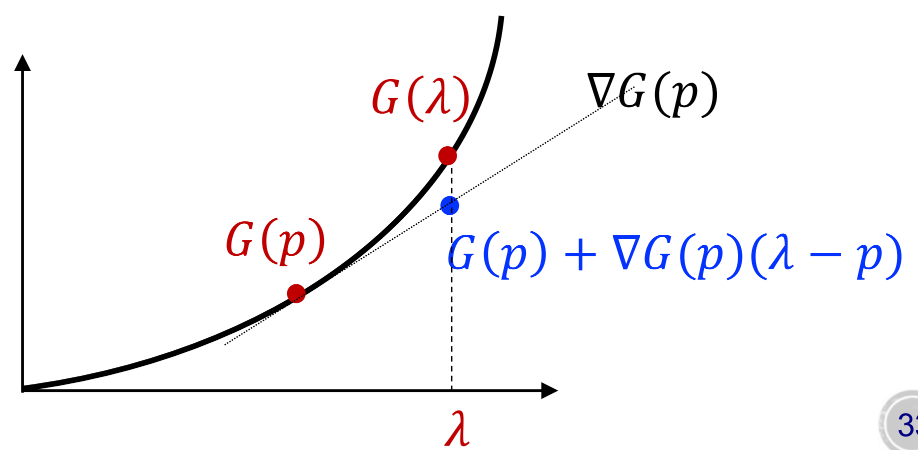
Proof of "⇐"

$$S(\lambda;p) = \mathbb{E}_{i\sim\lambda}[G(p) + \nabla G(p)(e_i - p)]$$

$$= G(p) + \nabla G(p)(\lambda - p)$$

$$\leq G(\lambda) = S(\lambda;\lambda)$$

By convexity



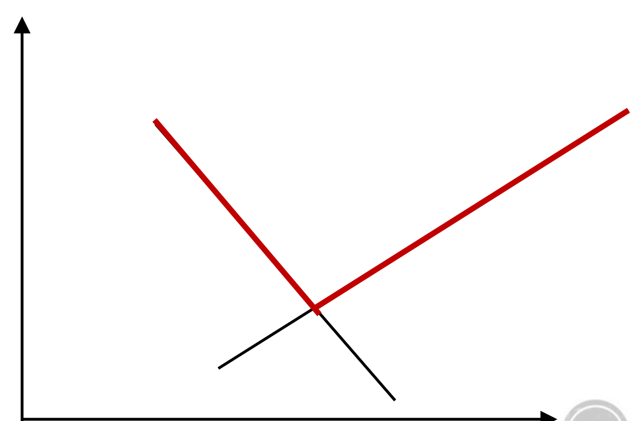
What $S(i; p)$ Are Proper?

Theorem. The scoring rule $S(i; p)$ is (strictly) proper if and only if there exists a (strictly) convex function $G: \Delta_n \to \mathbb{R}$ such that

$$S(i; p) = G(p) + \nabla G(p)(e_i - p)$$

Proof of "⇒"

- $S(\lambda; p) = \sum_{i \in [n]} \lambda_i S(i; p)$ is a linear fnc of λ for any p
- By properness, $S(\lambda; \lambda) = \max_{p \in \Delta_n} \sum_{i \in [n]} \lambda_i S(i; p)$, denoted as $G(\lambda)$
  - $G(\lambda)$ is convex in λ



What $S(i;p)$ Are Proper?

Theorem. The scoring rule $S(i;p)$ is (strictly) proper if and only if there exists a (strictly) convex function $G: \Delta_n \to \mathbb{R}$ such that

$$S(i;p) = G(p) + \nabla G(p)(e_i - p)$$

Proof of "⇒"

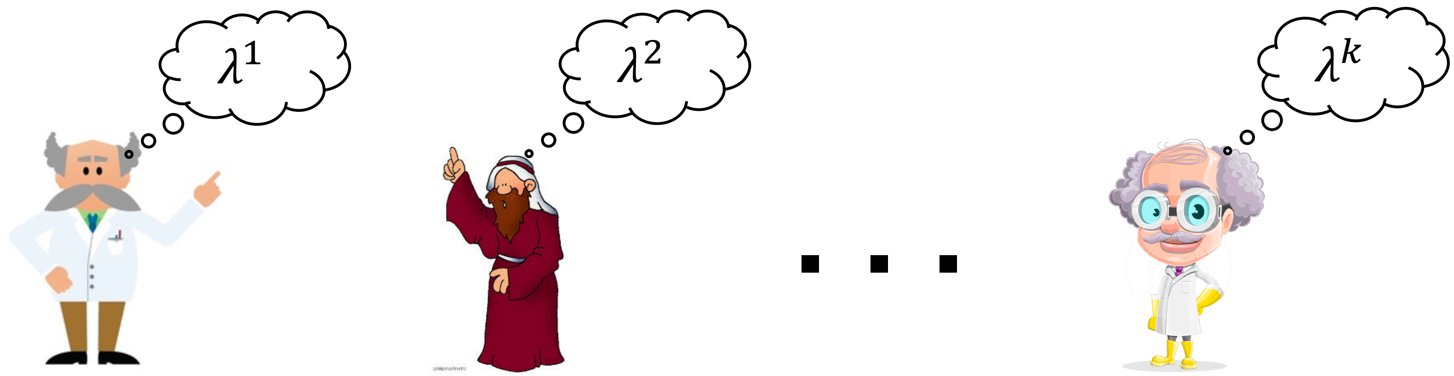
- $S(\lambda; p) = \sum_{i\in[n]} \lambda_i S(i;p)$ is a linear fnc of λ for any p
- By properness, $S(\lambda;\lambda) = \max_{p\in\Delta_n} \sum_{i\in[n]} \lambda_i S(i;p)$, denoted as $G(\lambda)$
  - $G(\lambda)$ is convex in λ
- The gradient of $G(\lambda)$ is the gradient of $\sum_{i\in[n]} \lambda_i S(i;p)$ for the $p = \lambda$
  - I.e., $\nabla G(\lambda) = S(\cdot\,;\lambda)$
- Thus,

$$\begin{aligned} S(i;p) &= S(p;p) + [S(i;p) - S(p;p)] \\ &= G(p) + S(\cdot;p)\cdot[e_i - p] \\ &= G(p) + \nabla G(p)[e_i - p] \end{aligned}$$

Outline

- Recap: Prediction Markets
- Scoring Rule and its Characterization
- Connection to Prediction Markets

What If There are Many Experts?



- One idea: elicit their predictions privately/separately
- Drawbacks
  1. May be expensive or wasteful – if experts all agree, we pay many times for the same prediction
  2. Not clear how to aggregate these predictions (average or geometric mean would not work)
  3. In fact, it may require experts' knowledge to correctly aggregate predictions

Sequential Elicitation

- Ask experts to make predictions in sequence
- The reward for expert k's prediction p^k will be

$$S(i; p^k) - S(i; p^{k-1})$$

  where p^{k-1} is the prediction of expert $k-1$
  - I.e., experts are paid based on how much they improved the prediction

Sequential Elicitation

- Ask experts to make predictions in sequence
- The reward for expert k's prediction p^k will be

$$S(i; p^k) - S(i; p^{k-1})$$

where p^{k-1} is the prediction of expert $k - 1$

  - I.e., experts are paid based on how much they improved the prediction

Theorem. If S is a proper scoring rule and each expert can only predict once, then each expert maximizes utility by reporting true belief given her own knowledge.

- Proof: since $S(i; p^{k-1})$ not under k's control, she maximizes reward by maximizing $S(i; p^k)$

Sequential Elicitation

- Ask experts to make predictions in sequence
- The reward for expert k's prediction p^k will be

$$S(i; p^k) - S(i; p^{k-1})$$

where p^{k-1} is the prediction of expert $k-1$

  - I.e., experts are paid based on how much they improved the prediction

Theorem. If S is a proper scoring rule and each expert can only predict once, then each expert maximizes utility by reporting true belief given her own knowledge.

Remarks:

- k may see previous reports and then update his prediction
  - Experts will aggregate predictions automatically

Sequential Elicitation

- Ask experts to make predictions in sequence
- The reward for expert k's prediction p^k will be

$$S(i; p^k) - S(i; p^{k-1})$$

where p^{k-1} is the prediction of expert $k-1$

  - I.e., experts are paid based on how much they improved the prediction

Theorem. If S is a proper scoring rule and each expert can only predict once, then each expert maximizes utility by reporting true belief given her own knowledge.

Remarks:

- Not true if an expert can predict for multiple times
  - She may manipulate her initial report to mislead others' prediction so that she has opportunity to significantly improve her prediction later
  - Will see an example in next lecture

Equivalence to Prediction Markets Described Previously

- It turns out that sequential elicitation is equivalent (in incentives) to the prediction market (PM) for buying and selling contracts
- Each expert moves the prediction to his own belief
  - Recall in PMs, expert will buy shares until prices hit his own belief
- Any strictly proper scoring rule can be used to implement a PM and any PM correspond to some proper scoring rules

Remarks

Mechanism design for prediction tasks

- ML is one way but not the only way of making predictions
- In some settings, aggregating predictions from experts is more desirable

Thank You

Haifeng Xu

University of Virginia

hx4ad@virginia.edu