

# CS243: Introduction to Algorithmic Game Theory

Crowdsourcing (Dengji ZHAO)

SIST, ShanghaiTech University, China

# Outline

- 1 Overview
- 2 Crowdsourcing Markets
- 3 Crowdsourcing Contests
- 4 Peer Prediction

# Motivations

- Imagine you need labels for 10,000 images. How can you tackle this problem?
- Imagine you need a new software architecture for your company's inventory management system. Due to the limited budget, is it possible for you to run an online contest instead of hiring a developer to design the new system.

## Crowdsourcing

Crowdsourcing describes the act of outsourcing a task or multiple tasks to a large, undefined group of workers via an open call.

# Motivations

- Imagine you need labels for 10,000 images. How can you tackle this problem?
- Imagine you need a new software architecture for your company's inventory management system. Due to the limited budget, is it possible for you to run an online contest instead of hiring a developer to design the new system.

## Crowdsourcing

Crowdsourcing describes the act of outsourcing a task or multiple tasks to a large, undefined group of workers via an open call.

# Examples

**Peer Production Systems** No employer-employee relationships. e.g., **Wikipedia**.

**Paid Crowdsourcing Markets** Some tasks, like labeling images, require payments to encourage contributions. e.g., **Amazon Mechanical Turk**.

**Crowdsourcing Contests** Workers are invited to compete to submit work in response to a job posting. e.g., **DARPA Red Balloon Challenge**.

# Outline

- 1 Overview
- 2 Crowdsourcing Markets**
- 3 Crowdsourcing Contests
- 4 Peer Prediction

# Examples: Amazon Mechanical Turk

amazon mechanical turk Artificial Intelligence

Your Account    HITS    Qualifications    270,761 HITS available now

Sign In

All HITS | HITS Available To You | HITS Assigned To You

Find  containing  that pay at least \$   For which you are qualified  require Master Qualification

**All HITS**  
1-10 of 4521 Results

Sort by: HIT Creation Date (newest first)     Show all details | Hide all details    1 2 3 4 5 > Next >> Last

<b>Product Search Relevance Side By Side - US (Closed)-Fri Mar 22 18:00:23 PDT 2013</b> <a href="#">View a HIT in this group</a>			
Requester: <a href="#">Amazon Requester Inc</a>	HIT Expiration Date: Apr 1, 2013 (1 week 2 days)	Reward: \$0.00	
	Time Allotted: 10 minutes	HITS Available: 2949	
<b>Copy Text from Business Card</b> <a href="#">View a HIT in this group</a>			
Requester: <a href="#">Oscar Smith</a>	HIT Expiration Date: Mar 22, 2013 (1 hour 59 minutes)	Reward: \$0.02	
	Time Allotted: 10 minutes	HITS Available: 1192	
<b>Copy Text from Business Card</b> <a href="#">View a HIT in this group</a>			
Requester: <a href="#">Oscar Smith</a>	HIT Expiration Date: Mar 22, 2013 (1 hour 59 minutes)	Reward: \$0.02	
	Time Allotted: 10 minutes	HITS Available: 1610	
<b>Easy User Interface Test: Button Clicking Game (LARGE BONUS GIVEN FOR CLICKING MORE IN THREE MINUTES!!! US only)</b> <a href="#">View a HIT in this group</a>			
Requester: <a href="#">HCTEST</a>	HIT Expiration Date: Mar 23, 2013 (23 hours 59 minutes)	Reward: \$0.03	
	Time Allotted: 2 hours	HITS Available: 5	
<b>Search: Keywords on Google.com (US)</b> <a href="#">View a HIT in this group</a>			
Requester: <a href="#">CrowdSource</a>	HIT Expiration Date: Mar 22, 2014 (52 weeks)	Reward: \$0.08	
	Time Allotted: 15 minutes	HITS Available: 14975	

# The Setting for Crowdsourcing Platforms

- There is one requester with budget  $B$ , and  $n$  workers arrive sequentially (in a random order).
- Each worker  $a_i$  has cost  $c_i \geq 0$  for each task and is willing to perform at most  $t_i \geq 0$  tasks.
- The worker's utility per task if paid price  $p_i$  is  $u_i = p_i - c_i$ . (Assume all tasks are of the same kind. e.g., labeling.)
- In every round, a new worker arrives and reports  $c_i$  and  $t_i$ . The requester decides how many tasks to allocate and the price per task.



# Pricing Mechanisms for Crowdsourcing Platforms

- Pay  $p = c_i$  per task: **Not truthful!**
- Pay a fixed price  $p^*$  per task: May be too high (wasting budget) or too low (no enough workers)!

## Target

Maximize the number of tasks that are completed without exceeding the budget and be strategy-proof.

# Dynamic Pricing

## Idea of Threshold Price

The mechanism calculates a **threshold price**  $p$  based on the reports from past workers, and then uses this price for a while (for next  $m$  workers, get allocated when cost is below  $p$  and pay  $p$ , otherwise get rejected), until the threshold price is updated again.

# Algorithm: Get Threshold

## An Intuition

What if the requester can know next  $m$  workers' type profile?

- If next  $m$  workers come with ascending costs ( $c_1 \leq c_2 \leq \dots \leq c_m$ ), then after each worker comes, the requester only to consider whether it is worth to improve the price (to allocate tasks to the new arrival worker).

# Algorithm: Get Threshold

## Question

What if the requester can know next  $m$  workers' type profile? (assuming  $c_1 \leq c_2 \leq \dots \leq c_m$ .) What's the price  $p$  for them?

- Let  $x_i$  be the number of tasks allocated to  $i$
- When worker 1 comes, it is always worth improving price to  $p = c_1$  to allocate  $x_i = t_i$ .
- When worker  $i$  comes, it is worth improving price to  $p = c_i$  if we can further allocate at least one more task to  $i$ . We shall check whether  $c_i(1 + \sum_{j < i} x_j) \leq B$  (we can afford the payments for first  $i$  agents when the price is  $c_i$ ).
- Allocate as much work as possible to worker  $i$ :

$$x_i = \min \left\{ t_i, \lfloor B/p \rfloor - \sum_{j < i} x_j \right\}$$

# Algorithm: Get Threshold

## Simulation

But the requester cannot know the type profile of next  $m$  workers. She can simulate it by assuming they have the same profile with past  $m$  workers.

**Input:** the past  $m$  workers' reports  $\{(c_1, t_1), \dots, (c_m, t_m)\}$ .

- 1 Sort reports such that  $c_1 \leq c_2 \leq \dots \leq c_m$ .
- 2 Set  $i = 1$ .
- 3 While  $c_i \leq B / (1 + \sum_{j < i} x_j)$  do:
  - 1  $p = c_i$
  - 2  $x_i = \min \{ t_i, \lfloor B/p \rfloor - \sum_{j < i} x_j \}$
  - 3  $i = i + 1$
- 4 Output  $p$  for the next  $m$  workers.

# Online Mechanism for Task Pricing

## Question

1. How to decide  $m$ ?
2. What's the price for first  $m$  workers?

## Solution

$m$  can also be determined dynamically!

# Online Mechanism for Task Pricing

## The Whole Pricing Mechanism (Singer and Mittal, 2011):

- 1 First set  $p_0 = \epsilon$  for the first worker.
- 2 Then we have the history of the first worker, set  $m = 1$  and we get a price  $p_1$  for the next worker.
- 3 Then we have the history of the first 2 workers, set  $m = 2$  and we get a price  $p_2$  for the next 2 workers.
- 4 Then we have the history of the first 4 workers, set  $m = 4$  and we get a price  $p_3$  for the next 4 workers.
- 5 ... ..
- 6 Then we have the history of the first  $2^k$  workers, set  $m = 2^k$  and we get a price  $p_{k+1}$  for the next  $2^k$  workers.
- 7 For each bucket of workers, set  $B_k = (2^{k-1}/n)B$ .

# Online Mechanism for Task Pricing

## Theorem

The online pricing mechanism is *budget feasible* and *strategy-proof*.



# What's More: Crowdsourcing Beyond Markets

- Sometimes your labour is freely used!

Jane


Last Name: Smith

Email: stopall

Pick your color:  
 Red  
 Green

Submit

Select all squares with street signs.



🔄 🎧 ⓘ

VERIFY

6345262

8001

Type the text

Privacy & Terms

reCAPTCHA™  
stop spam.  
read books.

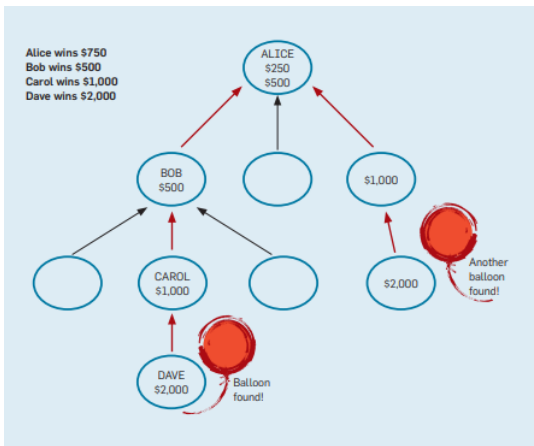
# Outline

- 1 Overview
- 2 Crowdsourcing Markets
- 3 Crowdsourcing Contests**
- 4 Peer Prediction

# Examples: DARPA Red Balloon Challenge



# Winning Solution in DARPA Red Balloon Challenge



# What's More: Sybil Attack

- If an agent pretends to be multiple agents to get more rewards, it is called Sybil attack.

## Question

What's the possible Sybil attack in the winning solution in DARPA Red Balloon Challenge?

# Outline

- 1 Overview
- 2 Crowdsourcing Markets
- 3 Crowdsourcing Contests
- 4 Peer Prediction**

# Prediction Markets

## Prediction Market

A prediction market is a financial market that is designed for information aggregation and prediction.

- Payoffs of the traded item is associated with outcomes of future events.

# Prediction Markets

Construct a prediction market to predict an uncertain event:

- 1 Turn an uncertain event of interest into a random variable.
- 2 Create a financial contract, and its payoff = value of the random variable.
- 3 Open a market in the financial contract and attract traders to wager and speculate (gambling).



# Prediction Markets

**Key Aspect:** payoff is uncertain!

In theory, price  $\approx$  (expectation of the random variable | all information hold by the market).

# Non-Market Alternative vs. Markets

- Ask Experts
  - Identifying experts can be hard
  - Incentives
  - Combining opinions can be difficult
- Prediction Markets
  - Self-selection
  - Monetary incentive and more
  - Real-time and self-organizing

# Non-Market Alternative vs. Markets

- Machine Learning

- Need huge historical data
- Past and future are related
- Hard to incorporate recent new information

- Prediction Markets

- No need for data
- No assumption on past and future
- Immediately incorporate new information

# Incentives for Experts: Proper Scoring Rules

- Report a probability estimate:  $r = \Pr\{r_1, r_2, \dots, r_n\}$ .
- Get payment  $s_i(r)$  if outcome  $\omega_i$  happens.
- Proper: incentive compatible
  - A risk neutral agent should choose  $r_i = \Pr(\omega_i)$  to maximize the expected profit.
- Proper scoring rules
  - Logarithmic:  $s_i(r) = a + b \log(r_i)$
  - Quadratic:  $s_i(r) = a + 2br_i - b \sum_j r_j^2$

# Prediction Market: Risk Management

- Why buying insurance?
  - If **something** is terrible to me, I buy a bunch of  
\$1 if *something* happens; \$0 otherwise
  - If **something** really happens, I am compensated.

# Prediction Market: Risk Management

- How insurance market works?
  - I am **risk averse** ( $u(x) = \log(x)$ ), insurance company is risk neutral ( $u(x) = x$ ). We both believe that *something* might happen with probability 0.01 and value  $x = 20000 - 10000 \times \mathbf{1}(\textit{something happens})$ .
  - My expected utility:

$$\mathbb{E}[u] = 0.01 \times \log(10000) + 0.99 \times \log(20000) \approx 4.2980$$

- My expected utility after buying \$10000 insurance for \$125:

$$\begin{aligned}\mathbb{E}[u] &= 0.01 \log(10000 + 10000 - 125) + 0.99 \log(20000 - 125) \\ &\approx 4.2983 > 4.2980\end{aligned}$$

- Expected utility of insurance company is  $0.01 \times (-9875) + 0.99 \times 125 = 25 > 0$
- Both insurance company and I are happy!

# Prediction Market: Risk Management

- Security Market
- Note, the insurance in the above example is in fact a contract

\$10000 if *something* happens; \$0 otherwise

- Market mechanism is to allocate risk and allow speculation among participants.

# Prediction Market: Examples

- Augur (www.augur.net)

The screenshot displays the 'Popular Markets' section of the Augur prediction market. It features a search bar at the top right and a navigation menu with 'OPEN (7)', 'IN-REPORTING', and 'RESOLVED'. The markets are sorted by 'MOST TRADED'. Each market entry includes its total volume, open interest, a warning for custom markets, the event title, expiration date, and a progress bar for 'Yes' and 'No' outcomes, along with an 'Invalid' button.

Market	Total Volume	Open Interest	Yes (%)	No (%)	Invalid (%)
Will Derek Chauvin be convicted of second-degree murder?	\$1,352	\$570	25.0%	0.00%	0.00%
Will the price of ETH/USD, exceed 10000 anytime between the open of August 6, 2020 and close of July 30, 2021, according to TradingView.com "ETHUSD (crypto - Coinbase)"?	\$1,510	\$1,510	10.0%	15.0%	0.00%
Lawful or unlawful, is Donald Trump the highest power of the United States of America, at 12:00 AM UTC on July 21st, 2021?	\$60	\$60	0.00%	52.7%	0.00%
Will Pope Francis vacate papacy by Jul 1, 2021 8:00 AM (GMT+8)?	\$0	\$0	0.00%	0.00%	0.00%
What price will KEEP/USD open at on July 31, 2021, according to TradingView.com for "KEEPUSD (crypto - Coinbase)"?	\$0				



# Prediction Market: Examples

## ● Successful results:

Total Dispute Stake 0.3494 REPV2	POLITICS / US POLITICS / PRESIDENT	Expired 8 JAN 2021
<b>Will Donald Trump win the 2020 U.S. Presidential election?</b>		
WINNING OUTCOME		
No		
Total Dispute Stake \$13,5507 REPV2	POLITICS / US POLITICS / PRESIDENT	Expired 21 DEC 2020
<b>Who will win the 2020 U.S. Presidential election?</b>		
WINNING OUTCOME		
Joe Biden		
Total Dispute Stake 0.3494 REPV2	CUSTOM MARKET- PROCEED WITH CAUTION  SCALAR / CRYPTO / ETHEREUM / DEFI	Expired 10 JAN 2021
<b>What Will the Total Value Locked (USD) In Defi Be on December 31, 2020?</b>		
WINNING OUTCOME		
14.8		
Total Dispute Stake 0.3494 REPV2	CUSTOM MARKET- PROCEED WITH CAUTION  CRYPTO / AUGUR	Expired 15 SEPT 2020
<b>Will There Be One Million Dollars Worth of Augur Liquidity on Balancer by September 15th, 2020</b>		
WINNING OUTCOME		
No		
Total Dispute Stake 0.3494 REPV2	CUSTOM MARKET- PROCEED WITH CAUTION  ECONOMICS / INDEX	Expired 1 OCT 2020
<b>Will the NASDAQ Composite Index close above 10,000.00 on September 30th, 2020?</b>		
WINNING OUTCOME		
Yes		
Total Dispute Stake 0.3494 REPV2	POLITICS / US POLITICS / U.S. SENATE	Expired 8 JAN 2021
<b>Which party will control the U.S. Senate after the 2020 election?</b>		
WINNING OUTCOME		
Democratic Party		
Total Dispute Stake 0.3494 REPV2	SPORTS / BASKETBALL / NBA	Expired 31 JUL 2020
<b>NBA: Will the LA Lakers win vs. the LA Clippers?</b> Estimated scheduled start time: Jul 31, 2020 9:00 AM (GMT+8)		
WINNING OUTCOME		
Yes		

# What's More: Open Questions

## 5 Open Questions in Prediction Markets (Wolfers and Zitzewitz, 2006)

- How to attract uninformed trader?
- How to tradeoff interest and contractability?
- How to limit manipulation?
- Are markets well calibrated on small probability?
- How to separate correlation from causation?

## Advanced Reading

- *Pricing Mechanisms for Crowdsourcing Markets*  
by Yaron Singer and Manas Mittal (WWW 2013)
- *Time-critical Social Mobilization*  
by G Pickard, W Pan, et al. (Science 2011)
- *Prediction Markets: Economics, Computation, and Mechanism Design*  
by Yiling Chen (EC-tutorial 2007)