Freie Universität Berlin

Prediction Markets
18.02.2014

Tawatchai Siripanya
Freie Universität Berlin

Instructors:
Prof. Dr. Wolfgang Mulzer
Yannik Stein

What is a prediction market?

- A mechanism designed to solved the information aggregation problem
- Are speculative markets aims to create predictions for the outcome of a future event
- It can use real or play money
- It encourages **market participants (traders)** to reveal their information (opinions and their confidence in those opinions) by **payoffs** associate with their contracts

Contracts

Prediction markets allow us (traders) to buy shares in an event

- If the event happens, we get $10 each share
- If the event does not happen, we get nothing
- Shares can be sold in advance of the event

Uncertainty, Risk, & Information

- Uncertainty e.g., rain **or** snow
- Risk e.g., Pr(rain) , Pr(snow)
- Information e.g., Pr(rain|info), Pr(snow|info)

Three outcomes

1€ iff

1€ iff

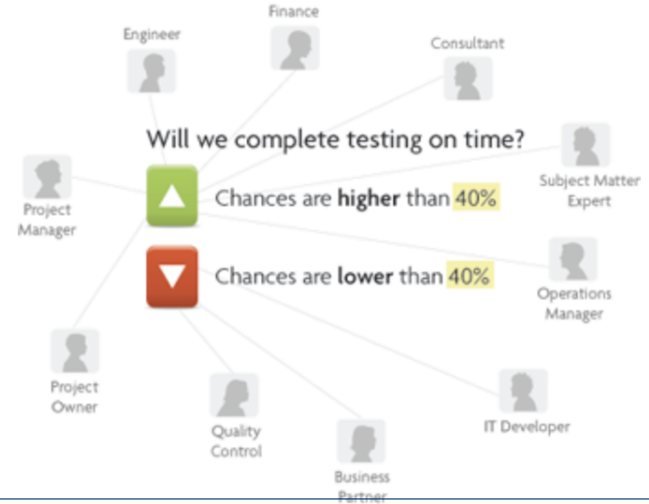
1€ iff

- If I think the probability this event will happen is **p=0.9**
  - **Buy** this contract at any price **less than** 0.9 Euro
  - **Sell** this contract at any price **greater than** 0.9 Euro

$\Longrightarrow$ Current price measures the population’s collective beliefs

Example: Inkling Markets

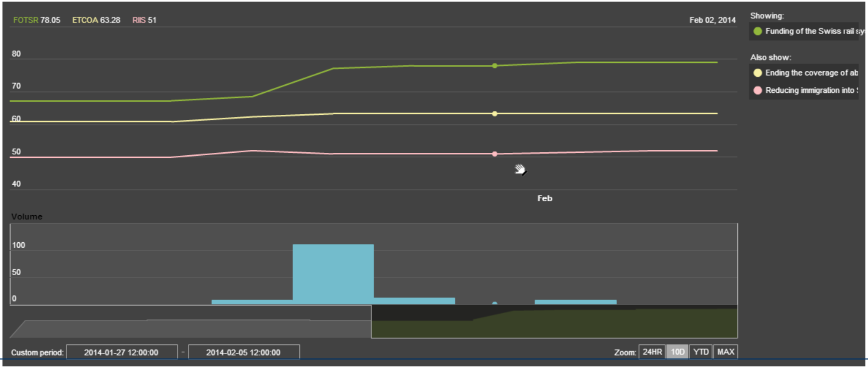
Example from [Abernethy2012]: Cooperate Prediction Markets



Example: Inkling Markets

Which of the three referendums will Swiss voters vote to approve on 9 February 2014?

- Ending the coverage of abortions
- Funding of the Swiss rail system
- Reducing immigration into Switzerland



Prediction Markets: Does it really work?

Yes, evidence from real markets

In Theory:

Under certain assumptions, prices converge to rational expectation equilibria, reflecting collective knowledge

In Practice:

- Election markets more accurate than polls [Berg2001][Pennock2002]
- HP internal markets beat sale forecast [Plott 2000]
- Orange juice futures beat FL weather forecasts [Roll1984]
- Sports betting markets provide accurate forecasts of game outcomes [Gandar1998][Thaler1988][DebnathEC'03][Schmidt2002]

List of Prediction Markets [Chen2008]

Real Money:

- Iowa Electronic Markets (IEM), http://www.biz.uiowa.edu/iem/
- InTrade, http://www.intrade.com
- Betfair, http://www.betfair.com/
- TradeSports, http://www.tradesports.com

Play Money:

- Hollywood Stock Exchange (HXS), http://www.hsx.com/
- Yahoo!/O'REILLY Tech Buzz Game, http://buzz.research.yahoo.com
- Inkling Markets http://inklingmarkets.com/

Internal Prediction Markets:

- HP, Google, Microsoft, Eli-Lilly, Corning . . .

How does prediction market work?

- Outcomes

1 Euro iff
horse **A** wins

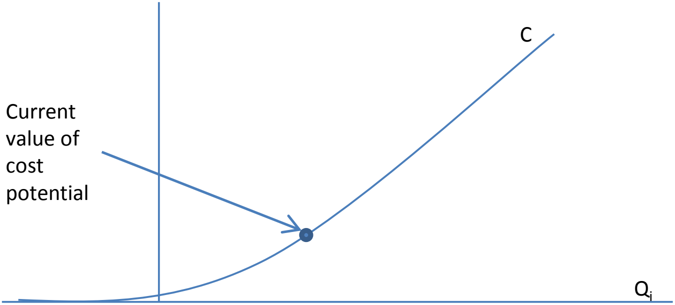
1 Euro iff
horse **B** wins

....

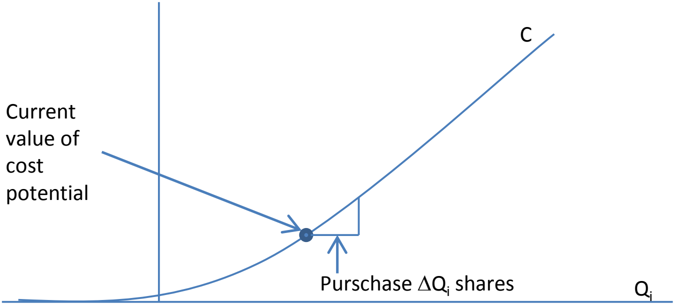
1 Euro iff
horse **N** wins

- Payments are determined by a cost potential function $C(\overrightarrow{q})$
- q_i is the current number of shares of the security for the out come i that have been purchased so far
- Current cost of buying a bundle $\overrightarrow{r}$ of shares is: $C(\overrightarrow{q}+\overrightarrow{r})-C(\overrightarrow{q})$

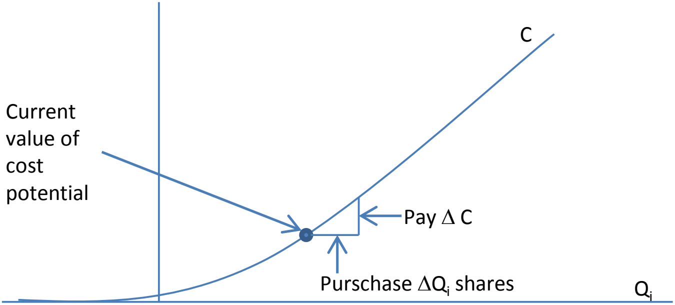
Cost Function



Cost Function

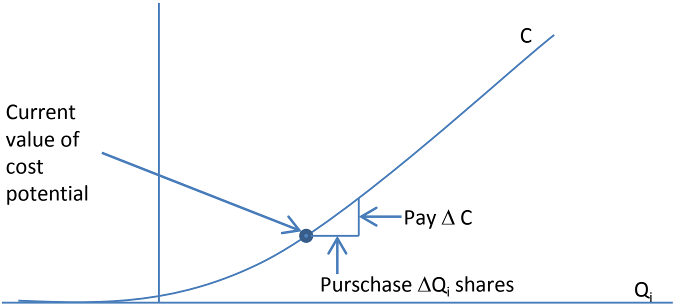


Cost Function



Cost Function

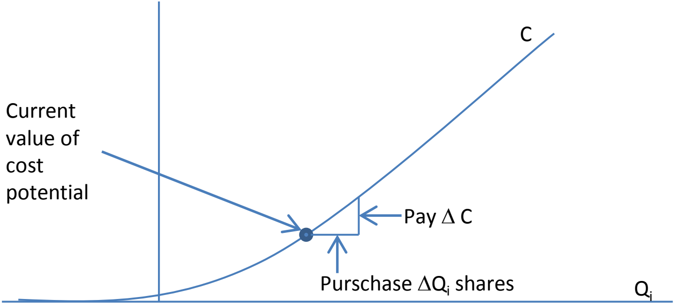
- Cost of purchase $= \Delta C$
- Unit price $= \Delta C / \Delta Q_i$
- Instantaneous price $= p_i = \delta C / \delta Q_i$



Cost Function

- Cost of purchase $= \Delta C$
- Unit price = $\Delta C / \Delta Q_i$
- Instantaneous price = $p_i = \delta C / \delta Q_i$

This is our prediction $\longrightarrow p_i = \delta C / \delta Q_i$



Logarithmic Market Scoring Rule (LMSR)

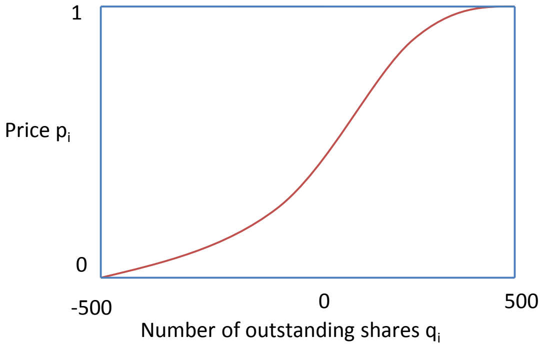
Logarithmic Market Scoring Rule (LMSR) [Hanson 2006]

- Cost function:

$$C(\vec{q}) = b\ ln\left(\sum_j e^{q_j/b}\right)$$

- Price function:

$$\partial C/\partial q_j = e^{q_j/b} / \sum_k e^{q_k/b}$$



Logarithmic Market Scoring Rule (LMSR) [Hanson 2006]

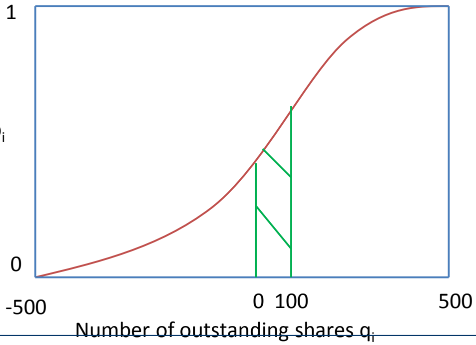
- Cost function:

$$C(\vec{q}) = b \; ln\left(\sum_j e^{q_j/b}\right)$$

- Price function:

$$\partial C/\partial q_j = e^{q_j/b} / \sum_k e^{q_k/b}$$

- Buy 100 shares



Logarithmic Market Scoring Rule (LMSR) [Hanson 2006]

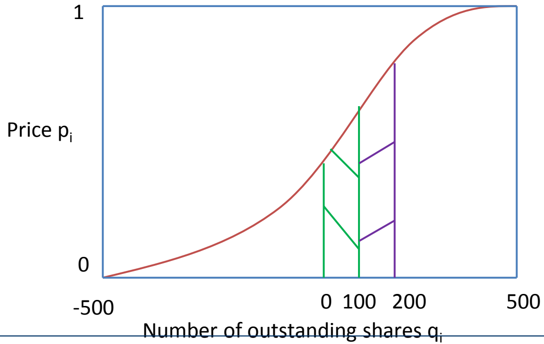
- Cost function:

$$C(\vec{q}) = b\ ln\left(\sum_j e^{q_j/b}\right)$$

- Price function:

$$\partial C/\partial q_j = e^{q_j/b} / \sum_k e^{q_k/b}$$

- Buy another 100 shares



Logarithmic Market Scoring Rule (LMSR) [Hanson 2006]

- Cost function:

$$C(\overrightarrow{q}) = b \; ln\left(\sum_j e^{q_j/b}\right)$$

- Price function:

$$\partial C/\partial q_j = e^{q_j/b} / \sum_k e^{q_k/b}$$

LMSR is "'reasonable"

- Price of security *i* increase with q_i
- Price sum to 1
- Predictable Market maker's loss : *b log N*

What about large outcomes?

n! 2^n infinite

Large state spaces–> LMSR is inefficient, price calculation becomes intractable

- Human is not good in addressing a small set of Pr()
- Independent outcomes, but ignore logical dependences

Machine learning/Statistics

- 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

Prediction Markets: Computational aspects

- Total number of possible securities
- Complexity of updating the price
- How many iterations before the price converge, if at all?

Example of auctioneer’s matching problem

- Given binary events A, B, and C
- We might bid to buy 5 units of a security $\langle A \wedge (B \vee C) \rangle$
- We will get $1 iif the event occurs
- Peter might bid to sell 10 units of a security $\langle A|C \rangle$
- Peter will get $1 iif the event occurs
- Conditional on event C occurring -> no payoff if C does not occur
- Bids maybe **divisible** (the bidders are willing to accept less than the requested quantity)
- Bids maybe **indivisible** (the bids must be fulfilled either completely or not at all)
- Given such bids, the auctioneer faces on matching problem

Complexity of Auctioneer's matching problem

- Let the length of the description of all the available securities be $O(n)$
- With n events,
  - The matching problem is polynomial in the **divisible** case (computationally feasible)
  - The matching problem is NP-complete in the **indivisible** case (intractable)
- With log n events,
  - The matching problem is co-NP-complete in the **divisible** case
  - The matching problem is $\sum_2^p$-complete in the **indivisible** case

- Definition: (Matching problem, divisible case) Given a set of orders O, does there exist $\alpha i \in [0, 1]$ with at least one $\alpha i > 0$ such that

$$For\ all\ \omega, \gamma^{\omega}_{auctionier} \geq 0$$

- Build a linear program
- We have variables α_i
- For each i, we have
- $0 \ \leq \alpha i \ \leq \ 1$
- And for each state ω in Ω we have the constraint
- $$\gamma^{\omega}_{auctionier} = \Sigma_i - \alpha_i \ \gamma^{\omega}_i >= 0$$
- Given these constraints we maximize $\Sigma_i \alpha_i$
- A set of orders has a matching exactly when $\Sigma_i \alpha_i > 0$

Show matching is in co-NP

- Let $m \leq n$ be the total number of buy and sell orders
- An upper bound on the objective function can be forced by a collection of $m+1$ constraints
- If **no matching exists** there must exist $m+1$constraints that force all the α_i to zero

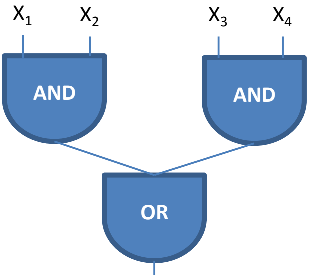
Show co-NP-completeness

- We reduce the NP-complete problem of Boolean formula satisfiability to the non existence of a matching
- Fix a formula ϕ
- Let the base securities be the variables of ϕ
- Consider the single security $\langle\phi\rangle$ with a buy order of 0.5
- If the formula ϕ is satisfiable then there is some state with payoff 0.5 (auctioneer payoff -0.5) and no fractional unit of the security $\langle\phi\rangle$ is a matching.
- If the formula ϕ is not satisfiable then every state has an auctioneer's payoff of 0.5 and a single unit of the security $\langle\phi\rangle$ is a matching

Distributed information markets

General formulation:

- Set up the market to compute some function $f(x_1, x_2, \ldots, x_n)$ of the information x_i available to each market participant (e.g., we want the market to compute future interest rates given other economic variables)



$$f(x_1, x_2, x_3, x_4) = (x_1 \wedge x_2) \vee (x_3 \wedge x_4)$$

Market Model: Security

- Each participant has some bit of information x_i
- The market aims at predicting the value of a Boolean function, $f(x) : \{0,1\}^n \longrightarrow \{0,1\}$
- One security is traded in the market. It pays:

$$= \begin{cases} 1 \; \textit{Euro} & \text{if } \text{f(X)=1} \\ 0 \; \textit{Euro} & \text{if } \text{f(x)=0} \end{cases}$$

Convergence time

- Let f be a weighted threshold function with n inputs,
- Let P be a prior probability distribution
- Then, after at most n rounds of trading, the price will reach its equilibrium value

$$p^{\infty} = f(x)$$

, where x denoted the combined information

Thank You

Q & A