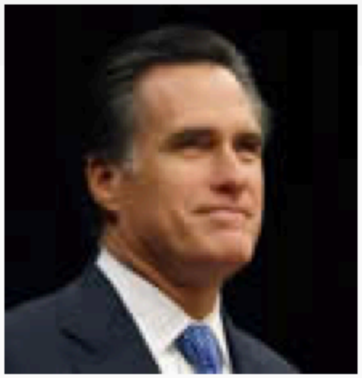
# Efficient Market Making Via Optimization & Connections to Online Learning

Jake Abernethy, Yiling Chen, and Jenn Wortman Vaughan

(Jenn's slides for CS269, Winter 2012)

# Information Markets



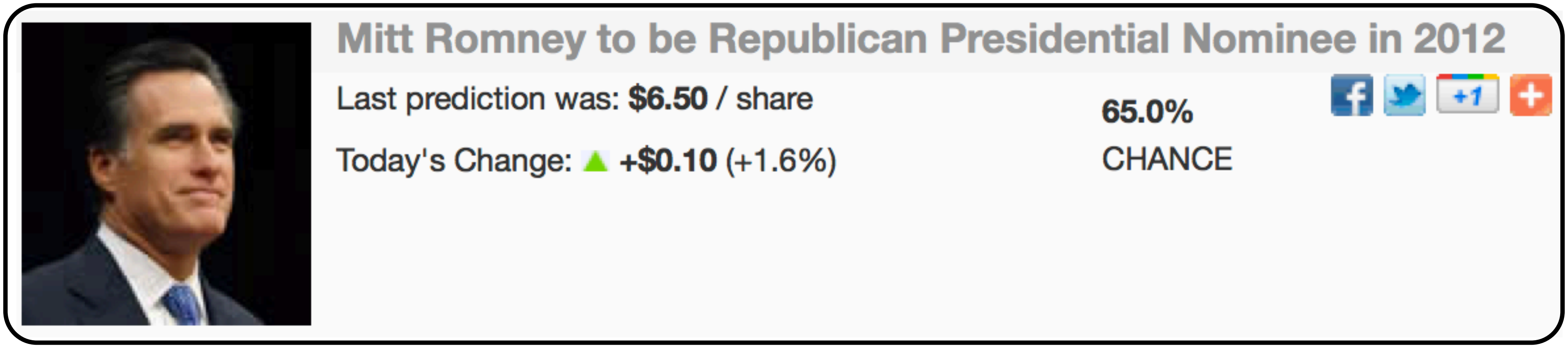
**Mitt Romney to be Republican Presidential Nominee in 2012**

Last prediction was: **$6.50** / share

Today's Change: ▲ **+$0.10** (+1.6%)

**65.0%**
CHANCE

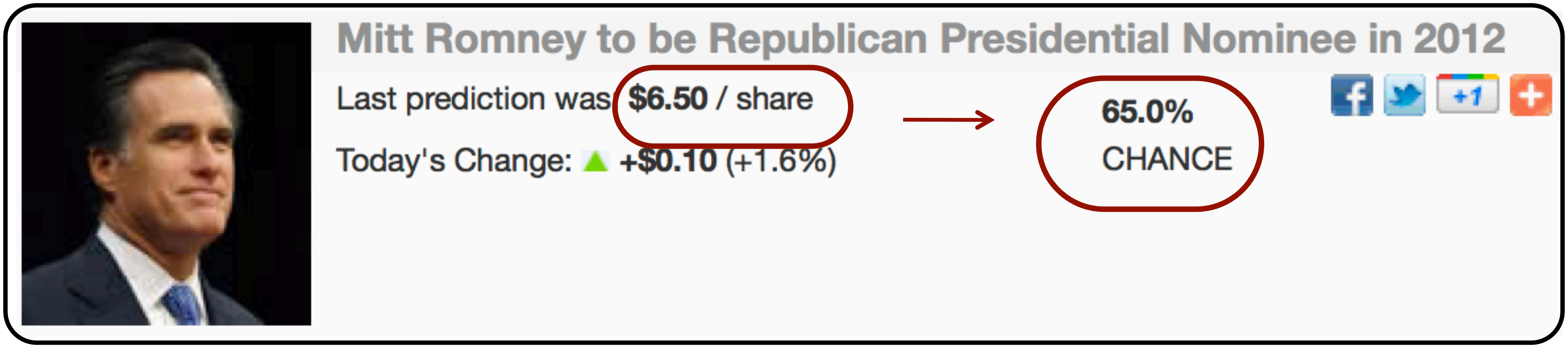
# Information Markets




Potential payoff is \$10. If I think that the probability of this event is $p$, I should

- Buy this security at any price less than \$10$p$
- Sell this security at any price greater than \$10$p$

# Information Markets




Potential payoff is \$10. If I think that the probability of this event is $p$, I should

- Buy this security at any price less than \$10$p$
- Sell this security at any price greater than \$10$p$

Current price measures the population's collective beliefs

# In case you're curious…

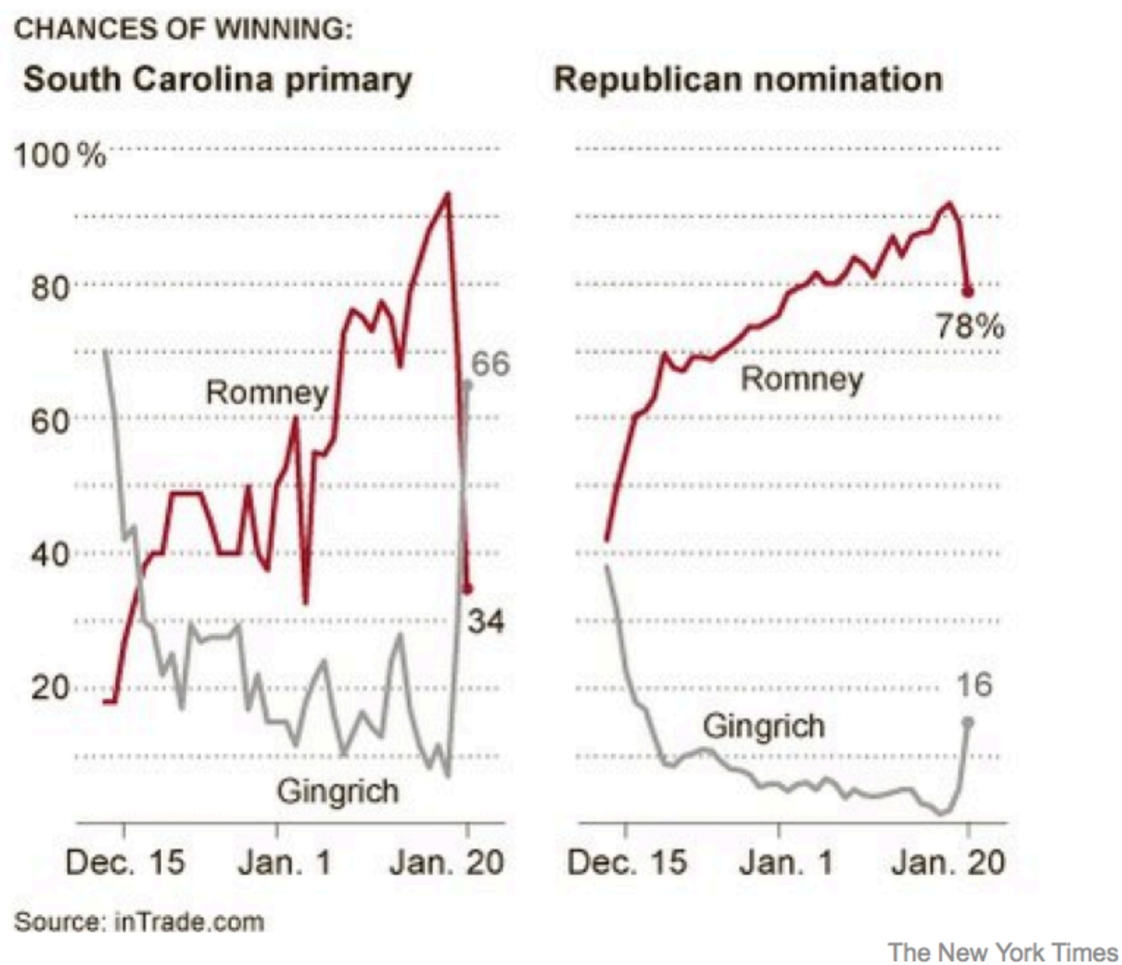
# The Caucus

The Politics and Government Blog of The Times


January 20, 2012, 8:21 PM

## Gingrich Gets a Boost on Intrade

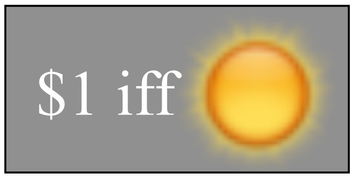
By RITCHIE S. KING




Newt Gingrich's presidential candidacy is on the upswing in South Carolina, if Intrade, a popular online market for political bets, is any indication.

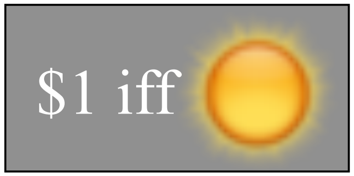
# Market Makers for Complete Markets

- In a complete market, a security is offered for each potential state of the world



# Market Makers for Complete Markets

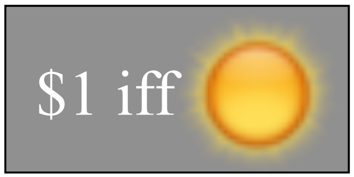
- In a complete market, a security is offered for each potential state of the world




- Market maker determines prices via a cost function $C$

# Market Makers for Complete Markets

- In a complete market, a security is offered for each potential state of the world

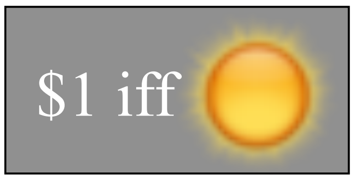



- Market maker determines prices via a cost function $C$
  - Let $q_i$ be the current number of shares of the security for state $i$ that have been purchased
  - Current cost of purchasing a bundle $\mathbf{r}$ of shares is

$$C(\mathbf{q} + \mathbf{r}) - C(\mathbf{q})$$

# Market Makers for Complete Markets

- In a complete market, a security is offered for each potential state of the world



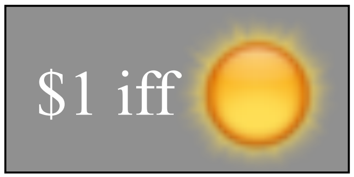

- Market maker determines prices via a cost function $C$
  - Let $q_i$ be the current number of shares of the security for state $i$ that have been purchased
  - Current cost of purchasing a bundle $\mathbf{r}$ of shares is

$$C(\mathbf{q} + \mathbf{r}) - C(\mathbf{q})$$

- Instantaneous price of security $i = \delta C \, / \, \delta q_i$

# Market Makers for Complete Markets

- In a complete market, a security is offered for each potential state of the world






- Market maker determines prices via a cost function $C$
  - Let $q_i$ be the current number of shares of the security for state $i$ that have been purchased
  - Current cost of purchasing a bundle $\mathbf{r}$ of shares is

$$C(\mathbf{q} + \mathbf{r}) - C(\mathbf{q})$$

- Instantaneous price of security $i = \delta C / \delta q_i$ "predictions"

# Example: LMSR

The Logarithmic Market Scoring Rule [Hanson, 2003] uses an exponential cost function

$$C(q_1,\ldots,q_N) = b\log \sum_{i=1}^{N} \exp(q_i/b)$$

and has instantaneous prices

$$p_i = \frac{\exp(q_i/b)}{\Sigma_j \exp(q_j/b)}$$

# Example: LMSR

The Logarithmic Market Scoring Rule [Hanson, 2003] uses an exponential cost function

$$C(q_1,\ldots,q_N) = b \log \sum_{i=1}^{N} \exp(q_i/b)$$

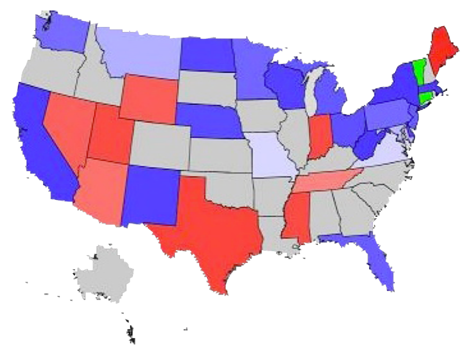
and has instantaneous prices

$$p_i = \frac{\exp(q_i/b)}{\Sigma_j \exp(q_j/b)}$$

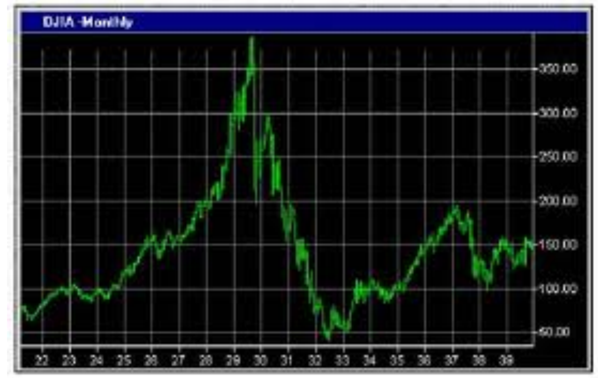
Notice that $p_i$ is increasing in $q_i$ and the prices sum to 1

# Complex Outcome Spaces


$n!$

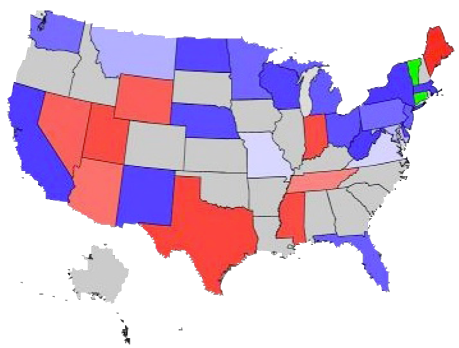


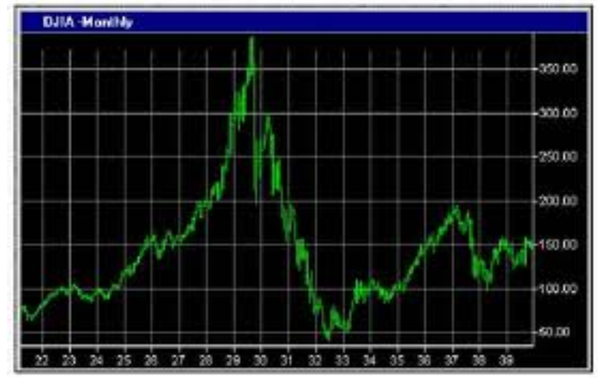
$2^n$




Infinite

# Complex Outcome Spaces

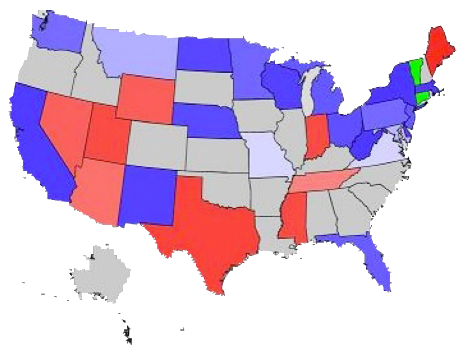



$n!$ $2^n$ Infinite

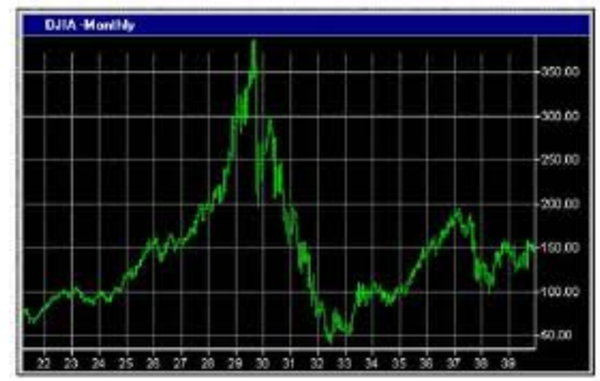
- Cannot simply run a standard market like LMSR
  - Calculating prices is intractable
  - Reasoning about probabilities is too hard for traders

# Complex Outcome Spaces


$n!$



$2^n$




Infinite

- Cannot simply run a standard market like LMSR
  - Calculating prices is intractable
  - Reasoning about probabilities is too hard for traders
- Can run separate, independent markets (e.g., horses to win, place, or show) but this ignores logical dependences

# Complex Outcome Spaces

**Our Goal:** Given a small set of securities over a very large (or infinite) state space, design a consistent market that can be operated efficiently

# Complex Outcome Spaces

**Our Goal:** Given a small set of securities over a very large (or infinite) state space, design a consistent market that can be operated efficiently

**Key Tools:** Convex optimization and conjugate duality

# Menu of Securities

We would like to offer a menu of securities $\{1, \ldots, K\}$ specified by a payoff function $\rho$

# Menu of Securities

We would like to offer a menu of securities $\{1, \ldots, K\}$ specified by a payoff function $\rho$

payoff

*securities*

*states*

| | | | | | |
|---|---|---|---|---|---|
| 10 | 0 | 5.5 | 0 | 17 | 0 |
| .9 | .9 | .9 | 0 | .9 | .9 |
| 0 | 42 | 0 | 10 | 10 | 10 |
| 0 | 0 | 11.5 | 8 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 |

# Example: Pair Betting

\$1 if and only if horse $i$ finishes ahead of horse $j$

# Example: Pair Betting

$1 if and only if horse *i* finishes ahead of horse *j*

| | A<B | B<A | A<C | C<A | B<C | C<B |
|---|---|---|---|---|---|---|
| ABC | 1 | 0 | 1 | 0 | 1 | 0 |
| ACB | 1 | 0 | 1 | 0 | 0 | 1 |
| BAC | 0 | 1 | 1 | 0 | 1 | 0 |
| BCA | 0 | 1 | 0 | 1 | 1 | 0 |
| CAB | 1 | 0 | 0 | 1 | 0 | 1 |
| CBA | 0 | 1 | 0 | 1 | 0 | 1 |

# What are “reasonable” prices?

# What are “reasonable” prices?

For complete markets…

# What are “reasonable” prices?

For complete markets… $\sum_i p_i = 1$

# What are "reasonable" prices?

For complete markets… $$\sum_i p_i = 1$$

For pair betting…

# What are "reasonable" prices?

For complete markets… $\sum_i p_i = 1$

For pair betting… $p_{i<j} + p_{j<i} = 1$

# What are "reasonable" prices?

For complete markets…

$$\sum_i p_i = 1$$

For pair betting…

$$p_{i<j} + p_{j<i} = 1$$

$$1 \le p_{i<j} + p_{j<k} + p_{k<i} \le 2$$

# What are “reasonable” prices?

For complete markets…

$$\sum_i p_i = 1$$

For pair betting…

$$p_{i<j} + p_{j<i} = 1$$

$$1 \le p_{i<j} + p_{j<k} + p_{k<i} \le 2$$

what else?

# What are "reasonable" prices?

For complete markets… $\sum_i p_i = 1$

For pair betting… $p_{i<j} + p_{j<i} = 1$

$$1 \le p_{i<j} + p_{j<k} + p_{k<i} \le 2$$

what else?

In general… ???

# An Axiomatic Approach

**Path independence:** The cost of acquiring a bundle **r** of securities must be the same no matter how the trader splits up the purchase.

# An Axiomatic Approach

**Path independence:** The cost of acquiring a bundle $\mathbf{r}$ of securities must be the same no matter how the trader splits up the purchase. Formally,

$$\text{Cost}(\mathbf{r} + \mathbf{r}' \mid \mathbf{r_1}, \mathbf{r_2}, \ldots, \mathbf{r_t})$$
$$= \text{Cost}(\mathbf{r} \mid \mathbf{r_1}, \mathbf{r_2}, \ldots, \mathbf{r_t}) + \text{Cost}(\mathbf{r}' \mid \mathbf{r_1}, \mathbf{r_2}, \ldots, \mathbf{r_t}, \mathbf{r})$$

# An Axiomatic Approach

**Path independence:** The cost of acquiring a bundle $\mathbf{r}$ of securities must be the same no matter how the trader splits up the purchase. Formally,

$$\begin{aligned}&\text{Cost}(\mathbf{r} + \mathbf{r}' \mid \mathbf{r_1}, \mathbf{r_2}, \ldots, \mathbf{r_t})\\ &\quad = \text{Cost}(\mathbf{r} \mid \mathbf{r_1}, \mathbf{r_2}, \ldots, \mathbf{r_t}) + \text{Cost}(\mathbf{r}' \mid \mathbf{r_1}, \mathbf{r_2}, \ldots, \mathbf{r_t}, \mathbf{r})\end{aligned}$$

This alone implies the existence of a cost potential function!

$$\begin{aligned}&\text{Cost}(\mathbf{r} \mid \mathbf{r_1}, \mathbf{r_2}, \ldots, \mathbf{r_t})\\ &\quad = C(\mathbf{r_1} + \mathbf{r_2} + \ldots + \mathbf{r_t} + \mathbf{r}) - C(\mathbf{r_1} + \mathbf{r_2} + \ldots + \mathbf{r_t})\end{aligned}$$

# An Axiomatic Approach

- **Existence of instantaneous prices:** $C$ must be continuous and differentiable

# An Axiomatic Approach

- **Existence of instantaneous prices:** $C$ must be continuous and differentiable

- **Information incorporation:** The purchase of a bundle **r** should never cause the price of **r** to decrease

# An Axiomatic Approach

- **Existence of instantaneous prices:** $C$ must be continuous and differentiable
- **Information incorporation:** The purchase of a bundle **r** should never cause the price of **r** to decrease
- **No arbitrage:** It is never possible to purchase a bundle **r** with a guaranteed positive profit regardless of outcome

# An Axiomatic Approach

- **Existence of instantaneous prices:** *C* must be continuous and differentiable
- **Information incorporation:** The purchase of a bundle **r** should never cause the price of **r** to decrease
- **No arbitrage:** It is never possible to purchase a bundle **r** with a guaranteed positive profit regardless of outcome
- **Expressiveness:** A trader must always be able to set the market prices to reflect his beliefs

# An Axiomatic Approach

**Theorem:** Under these five conditions, costs must be determined by a convex cost function $C$ such that

$$\{\nabla C(\mathbf{q}) : \mathbf{q} \in \mathrm{R}^K\} = \mathrm{Hull}(\boldsymbol{\rho})$$

# An Axiomatic Approach

**Theorem:** Under these five conditions, costs must be determined by a convex cost function $C$ such that

$$\boxed{\{\nabla C(\mathbf{q}) : \mathbf{q} \in \mathbb{R}^K\}} = \mathrm{Hull}(\boldsymbol{\rho})$$

reachable price vectors

# An Axiomatic Approach

**Theorem:** Under these five conditions, costs must be determined by a convex cost function $C$ such that

$$\boxed{\{\nabla C(\mathbf{q}) : \mathbf{q} \in \mathrm{R}^K\}} = \mathrm{Hull}(\boldsymbol{\rho})$$

reachable price vectors

*securities*

*states*

| | | | | | |
|---|---|---|---|---|---|
| 10 | 0 | 5.5 | 0 | 17 | 0 |
| .9 | .9 | .9 | 0 | .9 | .9 |
| 0 | 42 | 0 | 10 | 10 | 10 |
| 0 | 0 | 11.5 | 8 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 |

# An Axiomatic Approach

**Theorem:** Under these five conditions, costs must be determined by a convex cost function $C$ such that

$$\boxed{\{\nabla C(\mathbf{q}) : \mathbf{q} \in \mathrm{R}^K\}} = \mathrm{Hull}(\boldsymbol{\rho})$$

reachable price vectors

*securities*

*states*

| | | | | |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 |

# How do we find these cost functions?

# How do we find these cost functions?

- **Fact:** A closed, differentiable function $C$ is convex if and only if it can be written in the form

$$C(\mathbf{q}) = \sup_{\mathbf{x} \in \mathrm{dom}(R)} \mathbf{x} \cdot \mathbf{q} - R(\mathbf{x})$$

for a strictly convex function $R$ called the conjugate.

# How do we find these cost functions?

- **Fact:** A closed, differentiable function $C$ is convex if and only if it can be written in the form

$$C(\mathbf{q}) = \sup_{\mathbf{x} \in \mathrm{dom}(R)} \mathbf{x} \cdot \mathbf{q} - R(\mathbf{x})$$

for a strictly convex function $R$ called the conjugate.

Furthermore, $\nabla C(\mathbf{q}) = \arg\max_{\mathbf{x} \in \mathrm{dom}(R)} \mathbf{x} \cdot \mathbf{q} - R(\mathbf{x})$

# How do we find these cost functions?

- **Fact:** A closed, differentiable function $C$ is convex if and only if it can be written in the form

$$C(\mathbf{q}) = \sup_{\mathbf{x} \in \mathrm{dom}(R)} \mathbf{x}\cdot\mathbf{q} - R(\mathbf{x})$$

  for a strictly convex function $R$ called the conjugate.

  Furthermore, $\nabla C(\mathbf{q}) = \arg\max_{\mathbf{x} \in \mathrm{dom}(R)} \mathbf{x}\cdot\mathbf{q} - R(\mathbf{x})$

To generate a convex cost function $C$, we just have to choose an appropriate conjugate function and domain!

# But how do we choose $R$?

# But how do we choose $R$?

We can borrow ideas from online linear optimization, and in particular, Follow the Regularized Leader algorithms

- Our conjugate function ≈ their regularizer

Online Linear Opt.

Market Making

## Online Linear Opt.

- Learner maintains weights $\mathbf{w}_t \in K$ over $n$ items

## Market Making

## Online Linear Opt.

## Market Making

- Learner maintains weights $\mathbf{w}_t \in K$ over $n$ items
- Items have loss vector $\mathbf{l}_t$, cumulatively $\mathbf{L}_{t+1} = \mathbf{L}_t + \mathbf{l}_{t+1}$

## Online Linear Opt.

- Learner maintains weights $\mathbf{w}_t \in K$ over $n$ items
- Items have loss vector $\mathbf{l}_t$, cumulatively $\mathbf{L}_{t+1} = \mathbf{L}_t + \mathbf{l}_{t+1}$
- FTRL selects weights

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w} \in K} \mathbf{w} \cdot \mathbf{L}_t + \frac{1}{\eta} R(\mathbf{w})$$

## Market Making

## Online Linear Opt.

- Learner maintains weights $\mathbf{w}_t \in K$ over $n$ items
- Items have loss vector $\mathbf{l}_t$, cumulatively $\mathbf{L}_{t+1} = \mathbf{L}_t + \mathbf{l}_{t+1}$
- FTRL selects weights

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}\in K} \mathbf{w} \cdot \mathbf{L}_t + \frac{1}{\eta} R(\mathbf{w})$$

- Learner suffers regret

$$\sum_{t=1}^{T} \mathbf{w}_t \cdot \mathbf{l}_t - \min_{\mathbf{w}\in K} \mathbf{w} \cdot \mathbf{L}_T$$

## Market Making

## Online Linear Opt.

- Learner maintains weights $\mathbf{w}_t \in K$ over $n$ items
- Items have loss vector $\mathbf{l}_t$, cumulatively $\mathbf{L}_{t+1} = \mathbf{L}_t + \mathbf{l}_{t+1}$
- FTRL selects weights

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w} \in K} \mathbf{w} \cdot \mathbf{L}_t + \frac{1}{\eta} R(\mathbf{w})$$

- Learner suffers regret

$$\sum_{t=1}^{T} \mathbf{w}_t \cdot \mathbf{l}_t - \min_{\mathbf{w} \in K} \mathbf{w} \cdot \mathbf{L}_T$$

## Market Making

- Market maker maintains prices $\mathbf{x}_t \in \Pi$ over $n$ contracts

| <u>Online Linear Opt.</u> | <u>Market Making</u> |
|---|---|
| • Learner maintains weights $\mathbf{w}_t \in K$ over $n$ items | • Market maker maintains prices $\mathbf{x}_t \in \Pi$ over $n$ contracts |
| • Items have loss vector $\mathbf{l}_t$, cumulatively $\mathbf{L}_{t+1} = \mathbf{L}_t + \mathbf{l}_{t+1}$ | • Contracts are purchased in bundles $\mathbf{r}_t$, and $\mathbf{q}_{t+1} = \mathbf{q}_t + \mathbf{r}_{t+1}$ |
| • FTRL selects weights $$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}\in K} \mathbf{w} \cdot \mathbf{L}_t + \frac{1}{\eta} R(\mathbf{w})$$ | |
| • Learner suffers regret $$\sum_{t=1}^{T} \mathbf{w}_t \cdot \mathbf{l}_t - \min_{\mathbf{w}\in K} \mathbf{w} \cdot \mathbf{L}_T$$ | |

## Online Linear Opt.

- Learner maintains weights $\mathbf{w}_t \in K$ over $n$ items
- Items have loss vector $\mathbf{l}_t$, cumulatively $\mathbf{L}_{t+1} = \mathbf{L}_t + \mathbf{l}_{t+1}$
- FTRL selects weights

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w} \in K} \mathbf{w} \cdot \mathbf{L}_t + \frac{1}{\eta} R(\mathbf{w})$$

- Learner suffers regret

$$\sum_{t=1}^{T} \mathbf{w}_t \cdot \mathbf{l}_t - \min_{\mathbf{w} \in K} \mathbf{w} \cdot \mathbf{L}_T$$

## Market Making

- Market maker maintains prices $\mathbf{x}_t \in \Pi$ over $n$ contracts
- Contracts are purchased in bundles $\mathbf{r}_t$, and $\mathbf{q}_{t+1} = \mathbf{q}_t + \mathbf{r}_{t+1}$
- Market maker selects prices

$$\mathbf{x}_{t+1} = \arg\max_{\mathbf{x} \in \Pi} \mathbf{x} \cdot \mathbf{q}_t - R(\mathbf{x})$$

## Online Linear Opt.

- Learner maintains weights $\mathbf{w}_t \in K$ over $n$ items
- Items have loss vector $\mathbf{l}_t$, cumulatively $\mathbf{L}_{t+1} = \mathbf{L}_t + \mathbf{l}_{t+1}$
- FTRL selects weights

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}\in K} \mathbf{w} \cdot \mathbf{L}_t + \frac{1}{\eta} R(\mathbf{w})$$

- Learner suffers regret

$$\sum_{t=1}^{T} \mathbf{w}_t \cdot \mathbf{l}_t - \min_{\mathbf{w}\in K} \mathbf{w} \cdot \mathbf{L}_T$$

## Market Making

- Market maker maintains prices $\mathbf{x}_t \in \Pi$ over $n$ contracts
- Contracts are purchased in bundles $\mathbf{r}_t$, and $\mathbf{q}_{t+1} = \mathbf{q}_t + \mathbf{r}_{t+1}$
- Market maker selects prices

$$\mathbf{x}_{t+1} = \arg\max_{\mathbf{x}\in\Pi} \mathbf{x} \cdot \mathbf{q}_t - R(\mathbf{x})$$

- MM has worst-case loss

$$C(\mathbf{q}_0) - C(\mathbf{q}_T) + \max_{\mathbf{x}\in\Pi} \mathbf{x} \cdot \mathbf{q}_T$$

## Online Linear Opt.

- Learner maintains weights $\mathbf{w}_t \in K$ over $n$ items
- Items have loss vector $\mathbf{l}_t$, cumulatively $\mathbf{L}_{t+1} = \mathbf{L}_t + \mathbf{l}_{t+1}$
- FTRL selects weights

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}\in K} \mathbf{w} \cdot \mathbf{L}_t + \frac{1}{\eta} R(\mathbf{w})$$

- Learner suffers regret

$$\sum_{t=1}^{T} \mathbf{w}_t \cdot \mathbf{l}_t - \min_{\mathbf{w}\in K} \mathbf{w} \cdot \mathbf{L}_T$$

## Market Making

- Market maker maintains prices $\mathbf{x}_t \in \Pi$ over $n$ contracts
- Contracts are purchased in bundles $\mathbf{r}_t$, and $\mathbf{q}_{t+1} = \mathbf{q}_t + \mathbf{r}_{t+1}$
- Market maker selects prices

$$\mathbf{x}_{t+1} = \arg\max_{\mathbf{x}\in\Pi} \mathbf{x} \cdot \mathbf{q}_t - R(\mathbf{x})$$

- MM has worst-case loss

$$C(\mathbf{q}_0) - C(\mathbf{q}_T) + \max_{\mathbf{x}\in\Pi} \mathbf{x} \cdot \mathbf{q}_T$$

# More on Choosing *R*

- Interesting market properties can be described in terms of the conjugate…

# More on Choosing $R$

- Interesting market properties can be described in terms of the conjugate…
  - Worst-case market maker loss can be bounded by

$$\sup_{\mathbf{x} \in \mathrm{Hull}(\boldsymbol{\rho})} R(\mathbf{x}) - \inf_{\mathbf{x} \in \mathrm{Hull}(\boldsymbol{\rho})} R(\mathbf{x})$$

# More on Choosing $R$

- Interesting market properties can be described in terms of the conjugate…

  - Worst-case market maker loss can be bounded by

$$\sup_{\mathbf{x} \in \mathrm{Hull}(\boldsymbol{\rho})} R(\mathbf{x}) - \inf_{\mathbf{x} \in \mathrm{Hull}(\boldsymbol{\rho})} R(\mathbf{x})$$

  - Information loss (or the bid-ask spread, or the speed at which prices change) can be bounded too

# More on Choosing $R$

- Interesting market properties can be described in terms of the conjugate…
  - Worst-case market maker loss can be bounded by

$$\sup_{\mathbf{x} \in \mathrm{Hull}(\boldsymbol{\rho})} R(\mathbf{x}) - \inf_{\mathbf{x} \in \mathrm{Hull}(\boldsymbol{\rho})} R(\mathbf{x})$$

  - Information loss (or the bid-ask spread, or the speed at which prices change) can be bounded too

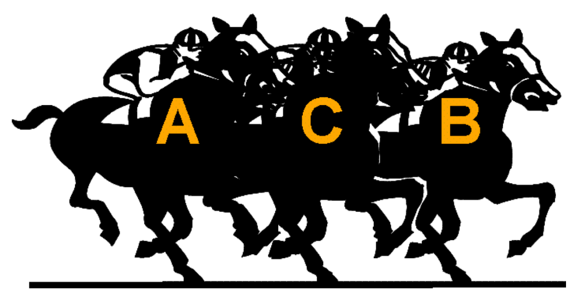
Gives us a way to optimize trade-offs in market design!

# Example: Permutations

- Suppose our state space is all permutations of $n$ items (e.g., candidates in an election, or horses in a race)

# Example: Permutations

- Suppose our state space is all permutations of $n$ items (e.g., candidates in an election, or horses in a race)
  - Pair bets: Bets on events of the form "horse $i$ finishes ahead of horse $j$" for any $i, j$
  - Subset bets: Bets on events of the form "horse $i$ finishes in position $j$" for any $i, j$



# Example: Permutations

- Suppose our state space is all permutations of $n$ items (e.g., candidates in an election, or horses in a race)
  - Pair bets: Bets on events of the form "horse $i$ finishes ahead of horse $j$" for any $i, j$
  - Subset bets: Bets on events of the form "horse $i$ finishes in position $j$" for any $i, j$


- Both known to be #P-hard to price using LMSR [Chen et al., 2008]

# Example: Permutations

- Suppose our state space is all permutations of $n$ items (e.g., candidates in an election, or horses in a race)
  - Pair bets: Bets on events of the form "horse $i$ finishes ahead of horse $j$" for any $i, j$
  - Subset bets: Bets on events of the form "horse $i$ finishes in position $j$" for any $i, j$


- Both known to be #P-hard to price using LMSR [Chen et al., 2008]
- Our framework handles both!

# Example: Permutations

Subset bets (“horse *i* finishes in position *j*”)

# Example: Permutations

Subset bets ("horse *i* finishes in position *j*")

- Hull($\boldsymbol{\rho}$) can be described by a small number of constraints:

$$\sum_j \text{price}(i \text{ in slot } j) = 1 \qquad \sum_i \text{price}(i \text{ in slot } j) = 1$$

# Example: Permutations

Subset bets ("horse *i* finishes in position *j*")

- Hull($\boldsymbol{\rho}$) can be described by a small number of constraints:

$$\sum_j \text{price}(i \text{ in slot } j) = 1 \qquad \sum_i \text{price}(i \text{ in slot } j) = 1$$

- Easily handled by our framework!

# Example: Permutations

Pair bets ("horse $i$ finishes ahead of horse $j$")

# Example: Permutations

Pair bets ("horse $i$ finishes ahead of horse $j$")

- Hull($\boldsymbol{\rho}$) is a bit uglier…

# Example: Permutations

Pair bets ("horse $i$ finishes ahead of horse $j$")

- Hull($\boldsymbol{\rho}$) is a bit uglier…
- Solution: Relax the no-arbitrage axiom

# Example: Permutations

Pair bets ("horse $i$ finishes ahead of horse $j$")

- Hull($\boldsymbol{\rho}$) is a bit uglier…
- Solution: Relax the no-arbitrage axiom
    - Allows us to to work with a larger, efficiently specified price space

# Example: Permutations

Pair bets ("horse *i* finishes ahead of horse *j*")

- Hull($\boldsymbol{\rho}$) is a bit uglier…
- Solution: Relax the no-arbitrage axiom
    - Allows us to to work with a larger, efficiently specified price space
    - But does it increase worst case loss?

# Example: Permutations

Pair bets ("horse $i$ finishes ahead of horse $j$")

- Hull($\boldsymbol{\rho}$) is a bit uglier…
- Solution: Relax the no-arbitrage axiom
  - Allows us to to work with a larger, efficiently specified price space
  - But does it increase worst case loss? No!

# Summary

- Our new optimization-based framework allows for the design of efficient market maker mechanisms for combinatorial or infinite state spaces

- Properties like worst-case loss and speed of price changes can be inferred easily

- Using this framework, we can design efficient markets for betting languages that are intractable to price using LMSR