Betting on Permutations

Brett Harrison

Harvard University

October 19, 2008

Introduction

- Last week, we looked at optimal strategies for traders
  - e.g. when to buy a security, what type of security to buy, how much of that security to buy
- Today, we look at optimal strategies for the auctioneer and the market maker
- "Betting on Permutations" focuses on the auctioneer's problem of risklessly match bets
- "Pricing Combinatorial Prediction Markets for Tournaments" will talk about the market maker's problem

Two Betting Languages

- Suppose we wish to bet on the finishing positions of n horses
  - There are $n!$ possible orderings of n horses, so theoretically $n!$ possible bets
  - Too expressive! Computationally hard to solve the auctioneer's problem
- The paper introduces two betting languages that restrict the number of available bets
  - Subset betting
  - Pair betting

Subset Betting

- Allows bets of the following two types:
  - $\langle \alpha | \Phi \rangle$: Pays \$1 if candidate α finishes in some position $\phi \in \Phi$, where Φ is non-empty set of positions
    - "Secretariat will finish in first, third, or fourth place."
  - $\langle \Psi | j \rangle$: Pays \$1 if some candidate $\psi \in \Psi$ finishes in position j, where Ψ is a non-empty set of horses
    - "Either Secretariat or Seabiscuit will finish in second place."
- Assuming there are n candidates and hence n positions, the total number of possible bets is $2^n + 2^n = 2^{n+1}$
- Even though the number of bets is exponential, the auctioneer can solve the matching problem in polynomial time!

Pair Betting

- Allows bets of the form $\langle \alpha > \beta \rangle$, which pays \$1 if candidate α finishes in a higher position than candidate β
- Assuming there are n candidates, the total number of possible bets is $n(n-1) = n^2 - n$
- Even though the number of bets is polynomial, the auctioneer's matching problem is NP-hard!

Matching Problems: Notation

- Let's define the auctioneer's riskless matching problem
- Notation:
  - n: number of candidates
  - $\mathcal{S}$: index set of states
  - $\mathcal{O}$: index set of bets/orders
  - $i \in \mathcal{O}$ is a triple (b_i, q_i, ϕ_i)
    - b_i is how much the trader is willing to pay for a unit share of security ϕ_i
    - q_i is the number of shares of ϕ_i the trader wants to purchase at price b_i
    - $b_i \in (0, 1)$, $q_i > 0$
  - $I_i(s)$: indicator variable; $I_i(s) = 1$ if the order is paid back \$1 in state s and $I_i(s) = 0$ otherwise
  - x_i: fraction of order $i \in \mathcal{O}$ accepted by the auctioneer

Matching Problems: Notation (continued)

- x_i: fraction of order $i \in \mathcal{O}$ accepted by the auctioneer
- Indivisible orders: $x_i \in \{0, 1\}$; auctioneer either accepts or rejects entire order
- Divisible orders: $x_i \in [0, 1]$; auctioneer can accept any fraction of the order

Problem Definitions

Definition (Existence of match, indivisible orders)

Given a set of orders $\mathcal{O}$, does there exist a set of $x_i \in \{0, 1\}$, $i \in \mathcal{O}$, with at least one $x_i = 1$ such that

$$\sum_i (b_i - I_i(s)) q_i x_i \geq 0, \quad \forall s \in \mathcal{S}?$$

Definition (Existence of match, divisible orders)

Given a set of orders $\mathcal{O}$, does there exist a set of $x_i \in [0, 1]$, $i \in \mathcal{O}$, with at least one $x_i > 0$ such that

$$\sum_i (b_i - I_i(s)) q_i x_i \geq 0, \quad \forall s \in \mathcal{S}?$$

Definition (Optimal match, indivisible orders)

Given a set of orders $\mathcal{O}$, choose $x_i \in \{0,1\}$ such that the following mixed integer programming problem achieves its optimality:

$$\begin{array}{lll} \max\limits_{x_i, c} & c & \\ \text{s.t.} & \sum_i (b_i - I_i(s)) q_i x_i \geq c, & \forall s \in \mathcal{S} \\ & x_i \in \{0,1\} & \forall i \in \mathcal{O}. \end{array}$$

Definition (Optimal match, divisible orders)

Given a set of orders $\mathcal{O}$, choose $x_i \in [0,1]$ such that the following linear programming problem achieves its optimality:

$$\begin{array}{lll} \max\limits_{x_i, c} & c & \\ \text{s.t.} & \sum_i (b_i - I_i(s)) q_i x_i \geq c, & \forall s \in \mathcal{S} \\ & 0 \leq x_i \leq 1, & \forall i \in \mathcal{O}. \end{array}$$

Complexity of Subset Betting

Theorem

The existence of a match and the optimal match problems with divisible orders in a subset betting market can both be solved in polynomial time.

Proof sketch.

The optimal match linear program can in polynomial time with the ellipsoid method if the corresponding separation problem can also be solved in polynomial time. This separation problem can be reduced from the maximum weighted bipartite matching problem, which can be solved in polynomial time. Furthermore, we can determine the existence of a match by observing whether the worst-case cost of the optimal match is positive. □

Let's break the proof down:

- Ellipsoid method: an algorithm for solving linear programs that has worse empirical performance than the simplex algorithm, but polynomial worst-case complexity in theory
- Separation problem: takes as input a vector of variable values and returns if the vector is feasible, otherwise returns a violated constraints
- In this case, asks whether there is a state (permutation) $s \in \mathcal{S}$ in which the profit is less than c, i.e.

$$\sum_i I_i(s) q_i x_i < \left(\sum_i b_i q_i x_i\right) - c \quad \forall s \in \mathcal{S}.$$

- This separation problem reduced from the maximum weighted bipartite matching problem...

- Let's construct a bipartite graph representation of this problem
  - Queue: Brett draws stuff on the board

Complexity of Pair Betting

Theorem

Computing the maximum worst-case profit in both indivisible and divisible pair betting NP-hard.

Proof sketch.

The indivisible and divisible orders problems reduce from the unweighted and weighted minimum feedback arc problems, respectively. □

- Construct a weighted graph $G = (V, E, w)$ where $V = \{1, \ldots, n\}$ and E contains an edge $e = (\alpha, \beta)$ with $w((\alpha, \beta)) = b_e$ for every order (α, β, b_e), i.e. for every order where the better will bet at most b_e that candidate α will beat candidate β.
  - Queue: Brett draws more

Theorem

Finding the optimal match in indivisible pair betting is NP-hard.

Proof sketch.

Again, problems reduce from minimum feedback arc problem: paper gives algorithms to answer the unweighted and weighted minimum feedback arc problems in polynomial time given oracles that provide the optimal matches in indivisible and divisible pair betting problems, respectively. □

Existence of a Match

Lemma

A sufficient condition for the existence of a match for pair betting is that there exists a cycle C in G such that

$$\sum_{e \in C} b_e \geq |C| - 1,$$

where $|C|$ is the number of edges in the cycle C.

Conclusion

- Subset betting and pair betting
- The order matching problem is easier for the more expressive betting language (subset betting)
  - What makes a betting language easy to work with? (What other problems can the order matching problem reduce from?)
  - What other betting languages are expressive but admit polynomial time algorithms for the corresponding matching problem?
- The bidder's order matching problem vs. the market maker's price setting problem