Combinatorial Betting

YILING CHEN
Yahoo! Research
LANCE FORTNOW
University of Chicago
EVDOKIA NIKOLOVA
Massachusetts Institute of Technology
and
DAVID M. PENNOCK
Yahoo! Research

We discuss the design of combinatorial betting mechanisms. We characterize the computational complexity of several variants of the problem and pose some oepn research questions.

Categories and Subject Descriptors: J.4 [**Computer Applications**]: Social and Behavioral Sciences – Economics

General Terms: Economics, Theory

Additional Key Words and Phrases: Prediction markets, expressive betting, order matching, computational complexity, mechanism design

1. INTRODUCTION

Committing to a bet is a credible way to state an opinion. Declaring that Hillary Clinton will win the 2008 US Presidential election is easy to say; staking real money on her victory at particular odds offers a quantifiable signal.

A *prediction market* is a betting intermediary designed to aggregate opinions about events of particular interest or importance. For example, intrade.com moderates bets on whether avian flu will hit the US in 2008. The market price reflects a stable consensus of a large number of opinions about the likelihood of an outbreak. Prediction market forecasts like those on intrade.com have a track record of remarkable accuracy.

Betting intermediaries abound, from Las Vegas to Wall Street, yet nearly all operate in a similar manner. Each allowable bet is explicitly listed and tracked; thus, by definition, the number of allowable bets is limited by available space. Each bet's outcome space is low dimensional: for example, "Hillary wins or loses" or "ACME stock price is x". Each bet type is managed independently, even when the bets are logically related: for example, stock options with different strike prices are traded in separate streams. Bets are processed either individually by a market maker or in pairs by a bilateral exchange, but almost never in groups of more than two at a time.

In this letter, we survey our attempts to design *combinatorial* betting mechanisms that support many more allowable bets and propagate information appropriately across logically-related bets. Thus, our mechanisms have the potential to both collect more information and process that information more fully than standard

mechanisms.

In a combinatorial betting system, allowable bets are not explicitly listed, but are rather implicitly defined on a combinatorial space. For example, the outcome space might be all $n!$ possible permutations of horses in a horse race, while the allowable bets are properties of permutations like "horse A finishes 3rd" or "horse A beats horses B and D". Or the outcome space might be all 2^{50} possible state-by-state results for the Democratic candidate in the 2008 US Presidential election, while allowable bets are Boolean statements like "Democrat wins in Ohio and Florida but not Texas".

The greater expressiveness comes at a computational cost for the betting intermediary or *auctioneer*, who now faces a difficult matching problem to connect up willing traders. In general, the auctioneer must look beyond bilateral matches to consider complex multilateral matches. Full expressiveness renders the matching problem intractable in general. However, in some cases we have discovered reasonable restrictions on expressiveness that admit polynomial-time matching algorithms.

Below we examine two different types of market combinatorics: (1) permutations and (2) Boolean combinations of binary events. In each case, we characterize the computational complexity of the auctioneer's matching problem under various degrees of expressiveness.

2. PERMUTATION BETTING

Consider wagering on the final ordering of n candidates; for example, the outcome of a horse race. The outcome space consists of all $n!$ possible permutations of candidates. Betting directly on complete orderings is both unnatural and intractable. Betting on *properties* of orderings (e.g., "candidate A wins", or "candidate B beats candidate D") is more natural though still intractable in most cases. We analyze two restricted betting languages—*pair betting* and *subset betting*—showing that the former remains intractable while the latter is tractable [Chen et al. 2007].

A pair bet is a bet on one candidate to finish ahead of another (e.g., "candidate A beats candidate B").

Subset bets come in two forms: position-subset bets and candidate-subset bets. A position-subset bet is a bet on a single candidate to finish in a subset of positions (e.g.,"candidate A finishes in position 1, 2, or 5"). A candidate-subset bet is a bet on a subset of candidates to finish in a single position (e.g., "candidate A, B, or D finishes in position 2").

Each bet defines a security that pays off \$1 per share if the specified event happens and \$0 otherwise. Participants place buy orders, specifying security, number of shares to buy, and the maximum price per share at which they are willing to buy.

After receiving all orders, a centralized auctioneer needs to solve an order matching problem to risklessly match up orders and decide what orders to accept and reject. Orders can be indivisible (must be accepted or rejected in full) or divisible (can be accepted partially). The auctioneer's matching problem can be formulated as an integer programming problem for indivisible orders and a linear programming problem for divisible orders, possibly with exponentially many constraints. We show that the auctioneer's matching problem is NP-hard for pair betting with both

divisible and indivisible orders. For subset betting with divisible orders, the corresponding separation problem of the linear programming formulation can be reduced to a maximum weighted bipartite matching problem, which has many polynomial-time solutions. Thus, the auctioneer's matching problem for subset betting with divisible orders can be solved in polynomial time using ellipsoid algorithm coupled with a maximum matching oracle.

3. BOOLEAN-STYLE BETTING

Boolean-style betting allows participants to bet on logical combinations of n binary events. The outcome space consists of all 2^n possible combinations of event outcomes. The allowable bets are Boolean formulas over the events, so there are 2^{2^n} allowable bets if we impose no restrictions. For example, events might be "Democrat wins Alabama", "Democrat wins Alaska", etc., for all fifty US states in the 2008 Presidential election. Allowable bets are logical propositions about the election outcome like "Democrat wins California and either Nevada or New Mexico".

Given a Boolean formula ϕ, the corresponding security pays off \$1 per share iff ϕ is true in the eventual outcome. More generally, we allow *conditional* securities $S_{\phi|\psi}$ based on two formulas ϕ and ψ; this is interpreted as, "make a payoff according to ϕ, conditional on ψ being true". In other words, the owner of security $S_{\phi|\psi}$ is paid \$1 per share if both ϕ and ψ are true, paid \$0 if ψ is true but ϕ is false, and the security is cancelled (and any money the owner paid for it is refunded) if ψ is false. Participants place orders that specify the security, number of shares to buy/sell, and the maximum/minimum price at which they are willing to buy/sell.

As before, the auctioneer's matching problem is to determine which orders to accept among all orders without incurring any risk. Fortnow et al. [2007] analyze the computational complexity of matching in Boolean-style betting. They show that for divisible orders the matching problem is computable in polynomial time when there are $O(\log m)$ events, but is co-NP-complete for $O(m)$ events, where m is the description length of all orders. For indivisible orders, the matching problem is even harder to solve. It's NP-complete for $O(\log m)$ events and Σ_2^p-complete for $O(m)$ events.[1] Indivisible order matching is hard even when bets are restricted to conjunctions of only two events.

4. FUTURE WORK AND OPEN QUESTIONS

We continue to search for useful betting languages that admit polynomial-time matching algorithms or approximations. In particular, for Boolean-style betting, we have not found any satisfactory betting language for the most interesting case of divisible orders and $O(m)$ events.

The matching problems discussed here are combinatorial analogs of traditional *call markets*. The auctioneer bears no risk. Combinatorial prediction markets divide participants' attention among an exponential number of outcomes, making the likelihood of finding agreeable trades low even with multilateral matching. This

[1] Indivisible order matching with $O(\log m)$ events can be thought of as analogous to the typical combinatorial auction problem discussed in this special issue, minus the inherent logical structure of the Boolean space.

may cause a *thin market* problem and ultimately a failure of information aggregation. Moreover, according to the so-called *no-trade theorems* in economics, rational traders shouldn't speculate at all with each other in a call market or any other zero-sum betting game.

Hence, it may be desirable to have a market maker subsidize trading and induce liquidity by offering prices for all allowable bets. A number of properties are desirable for combinatorial market makers including bounded loss, no arbitrage, tractability, unbounded shares offered, smooth pricing function, and modularity of price function. We are as yet unsure which properties are most important and which are achievable in combination under what circumstances. Also unknown is the relationship between the auctioneer matching problem and the market maker pricing problem.

REFERENCES

CHEN, Y., FORTNOW, L., NIKOLOVA, E., AND PENNOCK, D. M. 2007. Betting on permutations. In *ACM Conference on Electronic Commerce*. ACM Press, New York.

FORTNOW, L., KILIAN, J., PENNOCK, D. M., AND WELLMAN, M. P. 2004. Betting Boolean-style: A framework for trading in securities based on logical formulas. *Decision Support Systems 39,* 1, 87–104.