Pricing Combinatorial Markets for Tournaments

[Extended Abstract]

Yiling Chen
Yahoo! Research
111 West 40th Street
New York, NY 10018
cheny@yahoo-inc.com

Sharad Goel
Yahoo! Research
111 West 40th Street
New York, NY 10018
goel@yahoo-inc.com

David M. Pennock
Yahoo! Research
111 West 40th Street
New York, NY 10018
pennockd@yahoo-inc.com

ABSTRACT

In a prediction market, agents trade assets whose value is tied to a future event, for example the outcome of the next presidential election. Asset prices determine a probability distribution over the set of possible outcomes. Typically, the outcome space is small, allowing agents to directly trade in each outcome, and allowing a market maker to explicitly update asset prices. Combinatorial markets, in contrast, work to estimate a full joint distribution of dependent observations, in which case the outcome space grows exponentially. In this paper, we consider the problem of pricing combinatorial markets for single-elimination tournaments. With n competing teams, the outcome space is of size 2^{n-1}. We show that the general pricing problem for tournaments is #P-hard. We derive a polynomial-time algorithm for a restricted betting language based on a Bayesian network representation of the probability distribution. The language is fairly natural in the context of tournaments, allowing for example bets of the form "team i wins game k". We believe that our betting language is the first for combinatorial market makers that is both useful and tractable. We briefly discuss a heuristic approximation technique for the general case.

Categories and Subject Descriptors

F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity; G.3 [**Mathematics of Computation**]: Probability and Statistics

General Terms

Algorithms, Economics

Keywords

Bayesian networks, combinatorial markets, prediction markets, tournaments, logarithmic market scoring rule

1. INTRODUCTION

Every year in the NCAA Mens College Basketball playoff, 64 teams play in a single-elimination tournament known as "March Madness". In the first round, the 64 teams are paired into 32 games. The 32 first-round winners are then paired into 16 games in the second round, and so on, until one team emerges the winner of the championship game in the sixth and final round. There are 63 total games each with two outcomes and thus 2^{63} possible ways the tournament might unfold.

Millions of people try to predict the outcome of the tournament before it starts. Common office pool predictions take the form of categorical 0/1 statements of who will win in each of the 63 games. Las Vegas bookmakers and stock market style betting exchanges allow more informative probabilistic predictions. In fact, studies show that betting markets and related *prediction markets*, designed to forecast everything from elections to flu outbreaks, yield remarkably accurate probability estimates [1, 4, 12, 14, 15]. Even so, bets are typically limited to a relatively small number of predefined outcomes, for example the 32 opening round games and the championship game. Moreover, bets are managed independently so that, for example, a large bet on Duke to win the championship does not automatically update the odds of Duke winning in the first round.

We consider algorithms for implicitly maintaining a market-based probability distribution across all 2^{n-1} outcomes in an n team single-elimination tournament. In principle, this allows for betting on any of the $2^{2^{n-1}}$ possible properties of the tournament outcome, for example "Duke advances further than UNC", or "At least one of the top four teams loses in the first round". With a centralized market maker, information automatically propagates properly when logic dictates, for example ensuring that the probability of Duke advancing to round three is never greater than their probability of advancing to round two.

We show that the general pricing problem for tournaments is #P-hard. We then identify a restricted betting language that forms a tractable special case. We believe that the language is natural and useful, and the first such language for a combinatorial market maker. In our betting language, agents may buy and sell assets of the form "team i wins game k", and may also trade in conditional assets of the form "team i wins game k given that they reach that game" and "team i beats team j given that they face off". Although these are arguably natural bets to place, the expressiveness of the language has the surprising side effect of introducing dependencies between games which we would naively think to be independent. For example, it is possible in this language to have a market distribution in which the winners of first round games are not independent of one another. This phenomenon relates to results on the impossibility of preserving independence in an aggregate distribution [6, 13]. We show that the usual independence relationships are restored if we only permit bets of the form "team i beats team j given that they face off".

To prove our results, we represent market distributions as Bayesian networks, a well-studied structure for representing probability distributions. In typical applications, queries are made to the network to compute conditional probabilities under a fixed distribution. It is interesting to note that our algorithm uses the results of these queries to iteratively update the Bayesian network itself so as to mirror the evolving market distribution. A surprising feature of our representation is that network edges are necessarily oriented in the opposite direction suggested by the usual understanding of causality in tournaments. For example, instead of conditioning the distribution of second round games on the results of first round games, we condition on the results of third round games.

Related Work. Prior work on combinatorial markets has primarily focused on call market exchanges, in which agents place orders for assets, and the clearing problem is to risklessly match these orders between agents. Fortnow et al. [5] analyze, the computational complexity of Boolean-style betting, where the underlying outcome space is binary n-tuples and agents are allowed to bet on sets described by Boolean formulas. They show that for divisible orders the matching problem is co-NP-complete, and is $\sum_2^p$-complete for indivisible orders. Indivisible order matching is hard even when bets are restricted to conjunctions of only two literals. Chen et al. [3] analyze two languages for betting on permutations—pair betting and subset betting. 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". They show that subset betting is tractable while pair betting is not. Chen et al. [2]'s unpublished work relates most closely to our own. In both the Boolean and permutation market maker settings, they identify some very restrictive languages that remain #P-hard. They also propose an approximation algorithm for a permutation market maker.

Asset prices in the markets we analyze are determined by Hanson's logarithmic market scoring rule market maker [8], which has several advantages over call market designs. Agents trade with the market maker, who sets asset prices and who accepts all buy and sell orders at these prices. In particular, market makers help reduce both the *thin market* and *irrational participation* problems that affect stock market style exchanges. The thin market problem arises when agents have to coordinate which assets they will trade with each other, as is the case in call markets. The liquidity added by a market maker is especially important in supporting the large number of bet types typical in the combinatorial setting. In zero-sum games, 'no-trade' theorems [10] state that rational agents, after hedging their risks, will no longer trade with each other, even when they hold private information. Market-makers avoid this irrational participation issue by, in essence, subsidizing the market.

Our paper is organized as follows: In Section 2 we review the general framework of prediction markets and discuss Bayesian networks. In Section 3 we derive our main result, a polynomial-time algorithm to price combinatorial markets for single-elimination tournaments. Appendix A presents a heuristic approximation scheme for the general, #P-hard, problem of pricing combinatorial markets. The proofs of some of our results have been placed in Appendix B.

2. PRELIMINARIES

2.1 Prediction Markets

Prediction markets are speculative markets designed to elicit forecasts. Because the goal is to gather information, an automated market maker that expects to lose a small and bounded amount of money on average may actually help attract informed traders. In the next subsection, we describe the most common such market maker mechanism.

2.1.1 Market Scoring Rule Market Maker

A market scoring rule maintains a probability distribution over an outcome space Ω which reflects a consensus estimate of the likelihood of any event. Market scoring rules may be implemented as market-maker driven exchanges in which traders buy and sell securities of the form "Pays \$1 if ω occurs". All transaction costs are paid to a market-maker who agrees to honor the contracts. Let $q : \Omega \mapsto \mathbb{R}$ indicate the number of outstanding shares on each state. If a trader wishes to change the number of outstanding shares from q to $\tilde{q}$, i.e., wants to buy or sell shares, the cost of the transaction under the logarithmic market scoring rule [8] is $C(\tilde{q}) - C(q)$ where

$$C(q) = b \log \sum_{\tau \in \Omega} e^{q(\tau)/b}.$$

The parmeter b is the liquidity, or depth, of the market. When b is large, it becomes more expensive for any particular agent to move the market distribution. If there are q outstanding shares, the spot price for shares on a given outcome ω is

$$P_q(\omega) = \frac{d}{dq(\omega)} C(q) = \frac{e^{q(\omega)/b}}{\sum_{\tau \in \Omega} e^{q(\tau)/b}}$$

which is interpreted as the aggregate, market-generated probability estimate for ω.

Moving outstanding shares from q to $\tilde{q}$ results in a (state-dependent) net pay out to the agent of

$$\begin{aligned}
&[\tilde{q}(\omega) - q(\omega)] - [C(\tilde{q}) - C(q)] \\
&= \; b \log \left(\frac{e^{\tilde{q}(\omega)/b}}{\sum_{\tau \in \Omega} e^{\tilde{q}(\tau)/b}} \right) - b \log \left(\frac{e^{\tilde{q}(\omega)/b}}{\sum_{\tau \in \Omega} e^{\tilde{q}(\tau)/b}} \right) \\
&= \; b \log P_{\tilde{q}}(\omega) - b \log P_q(\omega).
\end{aligned}$$

In other words, for moving the prices from P_q to $P_{\tilde{q}}$, an agent receives payment $b \log P_{\tilde{q}}$ in exchange for agreeing to pay the last agent who interacted with the market.

The market-maker opens the market with an initial distribution of shares on states (e.g. $q_0 \equiv 0$). If the market closes with a distribution of shares $\tilde{q}$, the market-maker makes a net payment to traders of

$$[\tilde{q}(\omega) - q_0(\omega)] - [C(\tilde{q}) - C(q_0)] = b \log P_{\tilde{q}}(\omega) - b \log P_{q_0}(\omega) \\ \leq \; b \log(1/P_{q_0}(\omega)).$$

In particular, for $q_0 \equiv 0$, the market-maker's maximum loss is bounded by $b \log |\Omega|$.

2.1.2 Securities for Conditional Events

For an event $A \subset \Omega$, to construct the security "Pays \$1 if A occurs," a trader purchases one share on each outcome $\omega \in A$. Traders may also desire *conditional* securities of the form "Pays \$1 if A occurs, conditional on B occurring." If the condition B does not occur, then the transaction should be effectively voided. To construct this asset, traders buy shares of AB and short sell shares of $\bar{A}B$ for zero net payment (but assume liability in $\bar{A}B$). In this way, if B does not occur, the trader is not paid for AB, and does not have to cover shares in $\bar{A}B$. Otherwise, assuming B occurs, she is paid depending on whether A happens. Specifically, to simulate buying Δ shares of the security $A|B$, the agent buys

$$b \log \left(\frac{e^{\Delta/b}}{e^{\Delta/b} P_q(A|B) + P_q(\bar{A}|B)} \right) \tag{1}$$

shares of AB, and short sells

$$b \log \left(e^{\Delta/b} P_q(A|B) + P_q(\bar{A}|B) \right) \tag{2}$$

shares of $\bar{A}B$. Lemma B.1 shows that this transaction requires zero net payment. If $\bar{A}B$ occurs, the agent has to cover the shares she sold short, for a loss of dollars equal to (2), since each share pays \$1. In order to avoid extending credit to agents, the market-maker asks the agent to pay this potential loss up front, which is returned if $\bar{A}B$ does not occur. Then, if AB occurs, the agent receives:

$$b \log \left(\frac{e^{\Delta/b}}{e^{\Delta/b} P_q(A|B) + P_q(\bar{A}|B)} \right) \\ + b \log \left(e^{\Delta/b} P_q(A|B) + P_q(\bar{A}|B) \right) = \Delta.$$

If $\bar{A}B$ occurs, the agent receives nothing; and if B does not occur, the agent is returned her deposit of

$$b \log \left(e^{\Delta/b} P_q(A|B) + P_q(\bar{A}|B) \right).$$

2.2 Bayesian Networks

A Bayesian network (or a belief network) is a probabilistic graphical model that represents a set of variables and their probabilistic dependencies. Formally, a Bayesian network is a directed graph with labeled nodes corresponding to random variables $X_1, \ldots, X_n$, and edges drawn from lower to higher numbered nodes (see, e.g., Figures 1 and 2). The parents of a node X_i are those nodes that point to X_i. Given a joint distribution $P(X_1 = x_1, \ldots, X_n = x_n)$ on the nodes, a Bayesian network is a representation of P if

$$P(X_i = x_i | X_1, \ldots, X_{i-1}) = P(X_i = x_i | \text{parents}(X_i)).$$

These conditional probabilities, together with the structure of the Bayesian network, completely determine P. Namely,

$$\begin{aligned} &P(X_1 = x_1, \ldots, X_n = x_n) \\ &\quad = \prod_{i=1}^{n} P(X_i = x_i | X_1 = x_1, \ldots, X_{i-1} = x_{i-1}) \\ &\quad = \prod_{i=1}^{n} P(X_i = x_i | \text{parents}(X_i)). \end{aligned}$$

Although the Bayesian network completely specifies a distribution, in general it is NP-hard to compute, e.g., marginal probabilities $P(X_i = x_i)$. For certain network topologies, however, there exist efficient algorithms to compute both marginal and conditional distributions [11, 16]. In particular, for pricing tournaments, we rely on the fact that one can perform these computations on trees in time linear in the number of nodes.

3. PRICING COMBINATORIAL MARKETS FOR TOURNAMENTS

In Section 3.1 we present an elementary argument that the general pricing problem for tournaments is #P-hard. Given this difficulty, in Section 3.2 we derive a polynomial-time pricing algorithm by appropriately restricting to a natural betting language. The expressiveness of this language has the side effect of introducing dependencies between games which we would naively think to be independent, a phenomenon related to results on the impossibility of preserving independence in an aggregate distribution [6, 13]. In Section 3.3 we show that the usual independence relationships are restored if we further restrict the language.

3.1 Computational Complexity

The outcome space Ω for tournaments with n teams can be represented as the set of binary vectors of length $n-1$, where each coordinate denotes whether the winner of a game came from the left branch or the right branch of the tournament tree. Then $|\Omega| = 2^{n-1}$ and, in the most general version of the pricing problem, agents are allowed to bet on any of the $2^{2^{n-1}}$ subsets of Ω. The pricing problem is #P-hard, even under certain restrictions on the betting language.

LEMMA 3.1. *Suppose that there are no outstanding shares when the tournament market opens, and let ϕ be a Boolean formula. For $S_\phi = \{\omega : \omega$ satisfies $\phi\}$, $|S_\phi| = 2^{n-1}(e^{c/b} - 1)/(e^{1/b} - 1)$ where c is the cost of purchasing 1 share of S_ϕ and b is the liquidity parameter.*

PROOF. The cost of the transaction is

$$\begin{aligned} c &= b \log \left(|S_\phi| e^{1/b} + 2^{n-1} - |S_\phi| \right) - b \log 2^{n-1} \\ &= b \log \left(\frac{|S_\phi|}{2^{n-1}} \left(e^{1/b} - 1 \right) + 1 \right) \end{aligned}$$

and the result follows from solving for $|S_\phi|$. □

COROLLARY 3.1. *Suppose agents are allowed to place bets on sets S_ϕ where ϕ is a monotone 2-CNF, i.e., $\phi = c_1 \wedge \cdots \wedge c_r$ and c_i is the disjunction of 2 non-negated literals. Then the pricing problem restricted to this betting language is #P-hard.*

PROOF. The result follows from the fact that monotone #2-SAT is #P-complete [17]. □

3.2 A Tractable Betting Language

Given that the general pricing problem is #P-hard, we restrict the types of bets agents are allowed to place. Here we show how to support bets of the form "team i wins game k", "team i wins game k given that they make it to that game" and "team i beats team j given they face off." The key observation for pricing these assets is that bets in this language preserve the Bayesian network structure depicted in Figure 1, in which edges are directed away from the final game of the tournament. Surprisingly, these bets do not preserve the Bayesian structure corresponding to the usual understanding of causality in tournaments, in which arrows are reversed (see Figure 2).

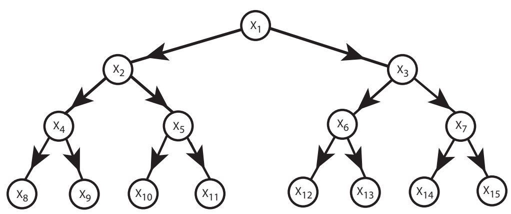

Figure 1: A Bayesian network for a tournament. Nodes represent game winners, and edges are oriented in reverse of that suggested by the usual notion of causality.

We start with some preliminary results. Lemma 3.2 and Corollary 3.2 show how, in an arbitrary market, probabilities are updated as the result of buying shares on an event. Lemma 3.3 shows how to simplify certain conditional probabilities for a Bayesian network structured as in Figure 1.

LEMMA 3.2. *Suppose Δb shares are purchased for the event A, where b is the liquidity parameter. Let P denote the distribution on Ω before the shares are purchased, and let $\tilde{P}$ denote the distribution after the purchase. Then for any event $B \subset \Omega$ we have*

$$\tilde{P}(B) = P(B)\left[\frac{e^{\Delta}P(A|B) + P(\bar{A}|B)}{e^{\Delta}P(A) + P(\bar{A})}\right].$$

PROOF. We use the notation q_ω to indicate the number of shares on outcome ω before the purchase, and $\tilde{q}_\omega$ for the number of shares after the purchase. Observe that

$$\begin{aligned}\frac{\tilde{P}(B)}{\tilde{P}(\bar{B})} &= \frac{\sum_{\omega\in AB} e^{\tilde{q}_\omega/b} + \sum_{\omega\in \bar{A}B} e^{\tilde{q}_\omega/b}}{\sum_{\omega\in A\bar{B}} e^{\tilde{q}_\omega/b} + \sum_{\omega\in \bar{A}\bar{B}} e^{\tilde{q}_\omega/b}} \\ &= \frac{e^{\Delta}\sum_{\omega\in AB} e^{q_\omega/b} + \sum_{\omega\in \bar{A}B} e^{q_\omega/b}}{e^{\Delta}\sum_{\omega\in A\bar{B}} e^{q_\omega/b} + \sum_{\omega\in \bar{A}\bar{B}} e^{q_\omega/b}} \\ &= \frac{e^{\Delta}P(AB) + P(\bar{A}B)}{e^{\Delta}P(A\bar{B}) + P(\bar{A}\bar{B})}.\end{aligned}$$

Now, since $\tilde{P}(B) = [\tilde{P}(B)/\tilde{P}(\bar{B})]/[1+\tilde{P}(B)/\tilde{P}(\bar{B})]$, we have

$$\begin{aligned}&\tilde{P}(B) = \\ &= \left[\frac{e^{\Delta}P(AB) + P(\bar{A}B)}{e^{\Delta}P(A\bar{B}) + P(\bar{A}\bar{B})}\right] \Big/ \left[1 + \frac{e^{\Delta}P(AB) + P(\bar{A}B)}{e^{\Delta}P(A\bar{B}) + P(\bar{A}\bar{B})}\right] \\ &= \frac{e^{\Delta}P(AB) + P(\bar{A}B)}{e^{\Delta}P(AB) + P(\bar{A}B) + e^{\Delta}P(A\bar{B}) + P(\bar{A}\bar{B})} \\ &= \frac{e^{\Delta}P(AB) + P(\bar{A}B)}{e^{\Delta}P(A) + P(\bar{A})} \\ &= P(B)\left[\frac{e^{\Delta}P(A|B) + P(\bar{A}|B)}{e^{\Delta}P(A) + P(\bar{A})}\right].\end{aligned}$$

□

COROLLARY 3.2. *Suppose Δb shares are purchased for the event A, where b is the liquidity parameter. Let P denote the distribution on Ω before the shares were purchased, and let $\tilde{P}$ denote the distribution after the purchase. Then for any events $B, C \subset \Omega$ we have*

$$\tilde{P}(B|C) = P(B|C)\left[\frac{e^{\Delta}P(A|BC) + P(\bar{A}|BC)}{e^{\Delta}P(A|C) + P(\bar{A}|C)}\right].$$

PROOF. The result follows from a Lemma 3.2 by writing $\tilde{P}(B|C) = \tilde{P}(BC)/\tilde{P}(C)$. □

LEMMA 3.3. *Consider a probability distribution P represented as a Bayesian network on a binary tree with arrows pointing away from the root and nodes labeled as in Figure 1. Select a node X_i with $i > 1$, and for $m < i$, let $X_{i,m}$ be the highest numbered node in $\{X_1, \ldots, X_m\}$ that lies along the unique path from the root to X_i. Then,*

$$P(X_i = x_i|X_1, \ldots, X_m) = P(X_i = x_i|X_{i,m}).$$

PROOF. Let $X_{i_0}, X_{i_1}, \ldots, X_{i_k}$ denote the path from X_i up to $X_{i,m}$, i.e. $X_{i_0} = X_i$, $X_{i_k} = X_{i,m}$ and X_{i_j} is the parent of $X_{i_{j-1}}$. By induction on k, the length of the path, we show

$$\begin{aligned}&P(X_i = x_i|X_1, \ldots, X_m) \\ &= \sum_{x_{i_1},\ldots,x_{i_{k-1}}} \Big[P(X_i = x_i|X_{i_1} = x_{i_1}) \\ &\qquad \times P(X_{i_1} = x_{i_1}|X_{i_2} = x_{i_2}) \cdots P(X_{i_{k-1}} = x_{i_{k-1}}|X_{i_k})\Big].\end{aligned}$$

For $k = 1$, $P(X_i = x_i|X_1, \ldots X_m) = P(X_i = x_i|X_{i_1})$, since $i_1 \leq m$ and, by the Bayesian network assumption, X_i is conditionally independent of its predecessors given its parent. For the inductive step, observe that

$$\begin{aligned}&P(X_i = x_i|X_1, \ldots, X_m) \\ &= \sum_{x_{i_1}} \Big[P(X_i = x_i|X_{i_1} = x_{i_1}, X_1, \ldots, X_m) \\ &\qquad \times P(X_{i_1} = x_{i_1}|X_1, \ldots, X_m)\Big] \\ &= \sum_{x_{i_1}} P(X_i = x_i|X_{i_1} = x_{i_1})P(X_{i_1} = x_{i_1}|X_1, \ldots, X_m).\end{aligned}$$

The result follows by applying the induction hypothesis to $P(X_{i_1} = x_{i_1}|X_1, \ldots, X_m)$. □

Theorem 3.1 and Corollary 3.3 show that bets on game winners preserve the Bayesian network structure, and, importantly, how to update the distribution.

THEOREM 3.1. *Suppose P is represented as a Bayesian network on a binary tree with nodes numbered as in Figure 1 and arrows pointing away from the root. Consider a market order $O = (g_j, t_j, \Delta b)$, interpreted as buying Δb shares on outcomes in which team t_j wins game g_j. Then the distribution $\tilde{P}$ that results from executing the order is also represented by a Bayesian network with the same structure, and only the distributions of g_j and its ancestors are affected. Furthermore, the uniform distribution P_0, corresponding to 0 shares on each outcome, is represented by the Bayesian network.*

PROOF. Each node X_i, excepting the root, has a unique parent which we call $\hat{X}_i$. We start by considering the uniform distribution P_0. Let $D(X_i)$ be the domain of X_i, i.e. $D(X_i) = \{t : \{X_i = t\} \neq \emptyset\}$. Observe that for each non-root node

$$P_0(X_i = x_i | X_1, \ldots, X_{i-1}) = \begin{cases} 1 & \hat{X}_i = x_i \text{ and } x_i \in D(X_i) \\ 1/|D(X_i)| & \hat{X}_i \notin D(X_i) \text{ and } x_i \in D(X_i) \\ 0 & \text{otherwise} \end{cases} .$$

In particular, $P_0(X_i = x_i | X_1, \ldots, X_{i-1}) = P_0(X_i = x_i | \hat{X}_i)$, showing that P_0 is represented by the Bayesian network.

Now we consider the case of updating. Set $A = \{X_{g_j} = t_j\}$, $B = \{X_i = x_i\}$ where X_i is a non-root node, $C = \{X_1 = x_1, \ldots, X_{i-1} = x_{i-1}\}$ for some configuration $(x_1, \ldots, x_{i-1})$, and $\hat{C} = \{\hat{X}_i = \hat{x}_i\}$ where $\hat{x}_i$ is the value of $\hat{X}_i$ in C. Corollary 3.2 shows that

$$\begin{aligned} \tilde{P}(B|C) &= P(B|C) \left[\frac{e^{\Delta} P(A|BC) + P(\bar{A}|BC)}{e^{\Delta} P(A|C) + P(\bar{A}|C)} \right] \\ &= P(B|\hat{C}) \left[\frac{e^{\Delta} P(A|BC) + P(\bar{A}|BC)}{e^{\Delta} P(A|C) + P(\bar{A}|C)} \right] \end{aligned}$$

where the last equality follows from the Bayesian network assumption. Consider the following cases:

1. $g_j < i$: Then, $P(A|BC) = P(A|C)$, and consequently, $\tilde{P}(B|C) = P(B|\hat{C})$.
2. $g_j = i$: Then, $P(A|BC) = P(A|B)$, and so
$$\tilde{P}(B|C) = P(B|\hat{C}) \left[\frac{e^{\Delta} P(A|B) + P(\bar{A}|B)}{e^{\Delta} P(A|\hat{C}) + P(\bar{A}|\hat{C})} \right].$$
3. $g_j > i$: In this case, Lemma 3.3 shows that
$$P(A|X_1, \ldots, X_i) = P(A|X_{g_j,i}).$$
If X_{g_j} is a descendent of X_i, then $X_{g_j,i} = X_i$, and $P(A|BC) = P(A|B)$, showing that
$$\tilde{P}(B|C) = P(B|\hat{C}) \left[\frac{e^{\Delta} P(A|B) + P(\bar{A}|B)}{e^{\Delta} P(A|\hat{C}) + P(\bar{A}|\hat{C})} \right].$$
Otherwise, if X_{g_j} is not a descendent of X_i, then
$$X_{g_j,i} \in \{X_1, \ldots, X_{i-1}\}$$
and hence $P(A|BC) = P(A|C)$. In this case, $\tilde{P}(B|C) = P(B|\hat{C})$.

In all three cases, we have that $\tilde{P}(B|C)$ depends only on the value of $\hat{X}_i$ and, in fact, only the distributions of g_j and its ancestors change, proving the result. □

COROLLARY 3.3. *Consider the setting of Theorem 3.1. The Bayesian network representing $\tilde{P}$ is constructed from the Bayesian network representing P as follows: For X_{g_j} and each of its ancestors, update the conditional probabilities according to*

$$\tilde{P}(X_i = x_i | \hat{X}_i = \hat{x}_i) = $$
$$= P(X_i = x_i | \hat{X}_i = \hat{x}_i) \left[\frac{(e^{\Delta} - 1) P(X_{g_j} = t_j | X_i = x_i) + 1}{(e^{\Delta} - 1) P(X_{g_j} = t_j | \hat{X}_i = \hat{x}_i) + 1} \right]$$

assuming X_i is not the root. Update the (unconditional) distribution of the root by

$$\tilde{P}(X_i = x_i) = $$
$$= P(X_i = x_i) \left[\frac{(e^{\Delta} - 1) P(X_{g_j} = t_j | X_i = x_i) + 1}{(e^{\Delta} - 1) P(X_{g_j} = t_j) + 1} \right].$$

The conditional distribution for all other nodes remain the same.

PROOF. The result follows from Theorem 3.1, Lemma 3.2 and Corollary 3.2 □

Above we showed how to update the market-based distribution on Ω as a result of market transactions. Lemma 3.4 shows how to compute the price of such a transaction.

LEMMA 3.4. *Suppose Δb shares are purchased for the event A, and let P denote the distribution on Ω before the shares are purchased. Then the cost of the purchase is*

$$b \log \left(e^{\Delta} P(A) + P(\bar{A}) \right).$$

To support conditional bets, we first show how to support bets in which agents pick the winners of two games, one of which is the parent game of the other. By combining these securities, one can construct the conditional assets as well.

THEOREM 3.2. *Suppose P is represented as a Bayesian network on a binary tree with nodes numbered as in Figure 1 and arrows pointing away from the root. Consider a market order $O = (g_{j_1}, t_{j_1}, g_{j_2}, t_{j_2}, \Delta b)$, interpreted as buying Δb shares on outcomes in which team t_{j_i} wins game g_{j_i}, where g_{j_1} is the parent of g_{j_2}. Then the distribution $\tilde{P}$ that results from executing the order is also represented by a Bayesian network with the same structure, and only the distributions of g_{j_2} and its ancestors are affected.*

PROOF. Each node X_i, excepting the root, has a unique parent which we call $\hat{X}_i$. Set $A = \{X_{g_{j_1}} = t_{j_1}, X_{g_{j_2}} = t_{j_2}\}$, $B = \{X_i = x_i\}$ where X_i is a non-root node, $C = \{X_1 = x_1, \ldots, X_{i-1} = x_{i-1}\}$ for some configuration $(x_1, \ldots, x_{i-1})$, and $\hat{C} = \{\hat{X}_i = \hat{x}_i\}$ where $\hat{x}_i$ is the value of $\hat{X}_i$ in C. Corollary 3.2 shows that

$$\begin{aligned} \tilde{P}(B|C) &= P(B|C) \left[\frac{e^{\Delta} P(A|BC) + P(\bar{A}|BC)}{e^{\Delta} P(A|C) + P(\bar{A}|C)} \right] \\ &= P(B|\hat{C}) \left[\frac{e^{\Delta} P(A|BC) + P(\bar{A}|BC)}{e^{\Delta} P(A|C) + P(\bar{A}|C)} \right] \end{aligned}$$

where the last equality follows from the Bayesian network assumption on P. Consider the following cases:

1. $g_{j_2} < i$: Then, $P(A|BC) = P(A|C)$, and consequently, $\tilde{P}(B|C) = P(B|\hat{C})$.

2. $g_{j_2} = i$: Then, $P(A|BC) = P(A|B\hat{C})$, and so

$$\tilde{P}(B|C) = P(B|\hat{C}) \left[\frac{e^{\Delta} P(A|B\hat{C}) + P(\bar{A}|B\hat{C})}{e^{\Delta} P(A|\hat{C}) + P(\bar{A}|\hat{C})} \right].$$

3. $g_{j_1} < i < g_{j_2}$: Set $A_1 = \{X_{g_{j_1}} = t_{j_1}\}$ and $A_2 = \{X_{g_{j_2}} = t_{j_2}\}$. If $A_1 \cap C = \emptyset$, then $P(A|BC) = 0 = P(A|C)$. Otherwise, by Lemma 3.3,

$$\begin{aligned} P(A|BC) &= P(A_2|BC) \\ &= P(A_2|A_1) = P(A_2|C) = P(A|C). \end{aligned}$$

Consequently, $\tilde{P}(B|C) = P(B|\hat{C})$.

4. $g_{j_1} = i$: In this case, again using Lemma 3.3, $P(A|BC) = P(A|B)$. So,

$$\tilde{P}(B|C) = P(B|\hat{C}) \left[\frac{e^{\Delta} P(A|B) + P(\bar{A}|B)}{e^{\Delta} P(A|\hat{C}) + P(\bar{A}|\hat{C})} \right].$$

5. $g_{j_1} > i$: Using the notation of Lemma 3.3, $P(A|BC) = P(A|X_{g_{j_1},i})$. If $X_{g_{j_1}}$ is a descendent of X_i, then

$$X_{g_{j_1},i} = X_i$$

Fand $P(A|BC) = P(A|B)$, showing that

$$\tilde{P}(B|C) = P(B|\hat{C}) \left[\frac{e^{\Delta} P(A|B) + P(\bar{A}|B)}{e^{\Delta} P(A|\hat{C}) + P(\bar{A}|\hat{C})} \right].$$

Otherwise, if $X_{g_{j_1}}$ is not a descendent of X_i, then $X_{g_{j_1},i} \in \{X_1, \ldots, X_{i-1}\}$, and hence

$$P(A|BC) = P(A|C).$$

In this case, $\tilde{P}(B|C) = P(B|\hat{C})$.

In all five cases, we have that $\tilde{P}(B|C)$ depends only on the value of $\hat{X}_i$ and, in fact, only the distributions of g_{j_2} and its ancestors change, proving the result. □

Corollary 3.4. *Consider the setting of Theorem 3.2. The Bayesian network representing $\tilde{P}$ is constructed from the Bayesian network representing P as follows: For $X_{g_{j_2}}$ and each of its ancestors, update the conditional probabilities according to*

$$\begin{aligned} &\tilde{P}(X_i = x_i|\hat{X}_i = \hat{x}_i) = \\ &= P(X_i = x_i|\hat{X}_i = \hat{x}_i) \\ &\times \left[\frac{(e^{\Delta} - 1) P(X_{g_{j_1}} = t_{j_1}, X_{g_{j_2}} = t_{j_2}|X_i = x_i, \hat{X}_i = \hat{x}_i) + 1}{(e^{\Delta} - 1) P(X_{g_{j_1}} = t_{j_1}, X_{g_{j_2}} = t_{j_2}|\hat{X}_i = \hat{x}_i) + 1} \right] \end{aligned}$$

assuming X_i is not the root. Update the (unconditional) distribution of the root by

$$\begin{aligned} &\tilde{P}(X_i = x_i) = \\ &= P(X_i = x_i) \\ &\quad \times \left[\frac{(e^{\Delta} - 1) P(X_{g_{j_1}} = t_{j_1}, X_{g_{j_2}} = t_{j_2}|X_i = x_i) + 1}{(e^{\Delta} - 1) P(X_{g_{j_1}} = t_{j_1}, X_{g_{j_2}} = t_{j_2}) + 1} \right]. \end{aligned}$$

The conditional distribution for all other nodes remain the same.

Proof. The result follows from Theorem 3.2, Lemma 3.2 and Corollary 3.2 □

To construct the conditional asset $A|B$, agents buy AB and short sell $\bar{A}B$ according to (1) and (2). In particular, to simulate "team i wins game k given that they make it to that game", set $A = \{X_k = i\}$ and $B = \{X_j = i\}$ where X_j is the child of X_k for which $B \neq \emptyset$. Theorem 3.2 directly shows how to update the Bayesian network after trading in AB. To execute $\bar{A}B$, one can trade, in sequence, the assets $A_1B, A_2B, \ldots, A_{i-1}B, A_{i+1}B, \ldots, A_nB$, where $A_l = \{X_k = l\}$. Now, Theorem 3.2 shows that each trade in A_lB preserves the Bayesian network, and furthermore, only the distributions of X_j and its ancestors are affected. Finally, knowing that $\bar{A}B$ preserves the network and that only X_j and its ancestors are affected, one need not actually trade each A_lB, but rather, may directly update the relevant distributions by appealing to Lemma 3.2 and Corollary 3.2.

Corollary 3.5. *Suppose P is represented as a Bayesian network on a binary tree with nodes numbered as in Figure 1 and arrows pointing away from the root. Set $A = \{X_k = i\}$ and $B = \{X_j = i\}$ where X_j is the child of X_k for which $B \neq \emptyset$. Then the distribution $\tilde{P}$ that results from buying Δb shares on $\bar{A}B$ is still represented by a Bayesian network with the same structure. Moreover, only the distributions of X_j and its ancestors are affected, and are updated as follows:*

$$\begin{aligned} &\tilde{P}(X_l = x_l|\hat{X}_l = \hat{x}_l) = \\ &\quad = P(X_l = x_l|\hat{X}_l = \hat{x}_l) \\ &\qquad \times \left[\frac{(e^{\Delta} - 1) P(X_k \neq i|X_j = i, \hat{X}_l = \hat{x}_l) + 1}{(e^{\Delta} - 1) P(X_k \neq i|\hat{X}_l = \hat{x}_l) + 1} \right] \end{aligned}$$

assuming X_l is not the root. Update the (unconditional) distribution of the root by

$$\tilde{P}(X_l = x_l) = P(X_l = x_l) \left[\frac{(e^{\Delta} - 1) P(X_k \neq i|X_j = i) + 1}{(e^{\Delta} - 1) P(X_k \neq i) + 1} \right].$$

The conditional distribution for all other nodes remain the same.

To construct the conditional asset "team i beats team j given that they face off" observe that there is a unique game k in which i and j could potentially play each other. Set $A = \{X_k = i\}$ and $B = \{X_{j_1} = i, X_{j_2} = j\}$ where X_{j_1} and X_{j_2} are the children of X_k ordered such that $B \neq \emptyset$. Now $AB = \{X_k = i, X_{j_2} = j\}$ and $\bar{A}B = \{X_k = j, X_{j_1} = i\}$. Theorem 3.2 allows agents to trade in both of these joint events, and they can consequently construct the conditional asset.

The price the market maker charges agents for buying these conditional assets is discussed in Section 2.1.2. Namely, the cost for purchasing Δb shares of $A|B$ is

$$b \log \left(e^{\Delta} P(A|B) + P(\bar{A}|B) \right).$$

Then, if AB occurs, the agent receive Δb dollars; if $\bar{A}B$ occurs, the agent receives nothing; and if B does not occur, the agent is returned the cost of the purchase.

Theorem 3.3. *For n teams, $O(n^3)$ operations are needed to update the Bayesian network as a result of trading assets of the form "team i wins game k", "team i wins game k given that they make it to that game" and "team i beats team j given they face off."*

PROOF. Each node in the k^{th} generation may take $n/2^k$ values with positive probability, where we set $k = 0$ for the root. The root maintains n marginal probabilities $P(X_1 = x_i)$. Each node in generation $k > 0$ maintains a conditional distribution $P(\,\cdot\,|\hat{X}_i = \hat{x}_i)$ for each of the $n/2^{k-1}$ values $\hat{x}_i$ its parent could take. If $\hat{x}_i$ is in the domain of X_i, then $P(X_i = \hat{x}_i|\hat{X}_i = \hat{x}_i) = 1$. Otherwise, specifying the conditional distribution of $X_i|\hat{X}_i = \hat{x}_i$ requires knowing $P(X_i = x_i|\hat{X}_i = \hat{x}_i)$ for each x_i in the domain of X_i. Consequently, X_i maintains $n/2^k \cdot n/2^k = n^2/4^k$ parameters. Trading in either conditional or unconditional assets affects the distribution of at most one node in each generation, and consequently changes $O(n^2)$ parameters. Since queries required to update the Bayesian network can be executed in time linear in the number of nodes [16], the total execution time for a trade is $O(n^3)$. □

3.3 Betting on matchup winners

The betting language discussed in Section 3.2 can lead to unexpected dependencies in the market-derived distribution. We illustrate this phenomenon with a simple example. Suppose there are four teams $\{T_1, \ldots, T_4\}$, so that the tournament consists of three games $\{X_1, X_2, X_3\}$, where X_2 and X_3 are the first round games, and X_1 is the final game. The state space Ω has eight outcomes

$$\begin{array}{llll} \omega_1 = (1,3,1) & \omega_2 = (1,3,3) & \omega_3 = (1,4,1) & \omega_4 = (1,4,4) \\ \omega_5 = (2,3,2) & \omega_6 = (2,3,3) & \omega_7 = (2,4,2) & \omega_8 = (2,4,4) \end{array}$$

where each coordinate indicates which team won the corresponding game.

Suppose we start with no outstanding shares, and are to execute two bets: "Δb shares on team 1 to win game 3" and "Δb shares on team 3 to win game 3". After executing these bets, outcomes ω_1, ω_2, ω_3 and ω_6 each have Δb shares, and the other outcomes have 0 shares. Now,

$$P(X_1 = 1) = P(X_2 = 3) = \frac{3e^{\Delta} + 1}{4e^{\Delta} + 4}$$

and $P(X_1 = 1, X_2 = 3) = 2e^{\Delta}/(4e^{\Delta} + 4)$. In particular, since $P(X_1 = 1)P(X_2 = 3) \neq P(X_1 = 1, X_2 = 3)$, X_1 and X_2 are not independent.

Here we further restrict the betting language of Section 3.2 so as to preserve the usual independence relations. The language allows only bets of the form "team i beats team j given that they face off." These bets preserve the Bayesian network structure shown in Figure 2. Notably, the edges in the network are directed toward the final game of the tournament, in constrast to the Bayesian network representing our more expressive language. In particular, the conditional distribution of a game X_j given all previous games depends only on the two games $\hat{X}_j^L$ and $\hat{X}_j^R$ directly leading up to X_j, as one might ordinarily expect to be the case.

THEOREM 3.4. *Suppose P is represented as a Bayesian network on a binary tree with nodes numbered as in Figure 2 and arrows pointing toward the root. Consider a market order $O = (g_j, t_j, t'_j, \Delta b)$, interpreted as buying Δb shares on outcomes in which team t_j wins game g_j, conditional on t_j and t'_j playing in game g_j. Then the distribution $\tilde{P}$ that results from executing the order is also represented by a Bayesian network with the same structure, and only the distribution of g_j is affected. Furthermore, the uniform distribution P_0, corresponding to 0 shares on each outcome, is represented by the Bayesian network.*

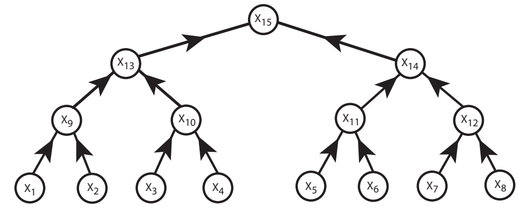

Figure 2: A Bayesian network for a tournament. Nodes represent game winners, and edges are oriented in accordance with the usual interpretation of causality.

COROLLARY 3.6. *Consider the setting of Theorem 3.4. The Bayesian network representing $\tilde{P}$ is constructed from the Bayesian network representing P as follows: For $A = \{X_{g_j} = t_j\}$ and $B = \{\{\hat{X}_{g_j}^L, \hat{X}_{g_j}^R\} = \{t_j, t'_j\}\}$, update the conditional probability $\tilde{P}(A|B)$ according to*

$$\tilde{P}(A|B) = \frac{e^{\Delta} P(A|B)}{e^{\Delta} P(A|B) + P(\bar{A}|B)}$$

(and set $\tilde{P}(\bar{A}|B) = 1 - \tilde{P}(A|B)$). All other conditional probabilities remain unchanged.

Every pair of teams play each other in at most one game, namely in the game that is their nearest common descendent in the tournament tree. Corollary 3.6 shows that one can think of this betting language as maintaining $\binom{n}{2}$ independent markets, one for each pair of teams, where each market gives an estimate of a particular team winning given they face off. Although bets in one market do not affect prices in any other market, they do effect the global distribution on Ω. In particular, the distribution on Ω is constructed from the independent markets via the Bayesian network.

Since each trade in this language requires updating only a single parameter of the Bayesian network, and since that update can be performed in $O(n)$ steps [16], the execution time for trades is linear in the number of teams.

Acknowledgments

We thank Robin Hanson and Daniel Reeves for helpful discussions.

4. REFERENCES

[1] J. E. Berg, R. Forsythe, F. D. Nelson, and T. A. Rietz. Results from a dozen years of election futures markets research. In C. A. Plott and V. Smith, editors, *Handbook of Experimental Economic Results (forthcoming)*. 2001.

[2] Y. Chen, L. Fortnow, N. Lambert, D. M. Pennock, and J. Wortman. Complexity of combinatorial market makers. Unpublished manuscript, arXiv:0802.1362v1 [cs.GT].

[3] Y. Chen, L. Fortnow, E. Nikolova, and D. M. Pennock. Betting on permutations. In *Proceedings of*

the Eighth ACM Conference on Electronic Commerce (EC'07)*, New York, June 2007. ACM.

[4] R. Forsythe, T. A. Rietz, and T. W. Ross. Wishes, expectations, and actions: A survey on price formation in election stock markets. *Journal of Economic Behavior and Organization*, 39:83–110, 1999.

[5] L. Fortnow, J. Kilian, D. M. Pennock, and M. P. Wellman. Betting boolean-style: A framework for trading in securities based on logical formulas. *Decision Support Systems*, 39(1):87–104, 2004.

[6] C. Genest and C. G. Wagner. Further evidence against independence preservation in expert judgement synthesis. *Aequationes Mathematicae*, 32(1):74–86, 1987.

[7] R. D. Hanson. Combinatorial information market design. *Information Systems Frontiers*, 5(1):107–119, 2003.

[8] R. D. Hanson. Logarithmic market scoring rules for modular combinatorial information aggregation. *Journal of Prediction Markets*, 1(1):1–15, 2007.

[9] A. Marshall. The use of multi-stage sampling schemes in monte carlo computations. In M. Meyer, editor, *Symposium on Monte Carlo Methods*, pages 123–140. Wiley, New York, 1956.

[10] P. Milgrom and N. L. Stokey. Information, trade and common knowledge. *Journal of Economic Theory*, 26(1):17–27, 1982.

[11] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishing, 1988.

[12] D. M. Pennock, S. Lawrence, C. L. Giles, and F. A. Nielsen. The real power of artificial markets. *Science*, 291:987–988, February 2002.

[13] D. M. Pennock and M. P. Wellman. Graphical models for groups: Belief aggregation and risk sharing. *Decision Analysis*, 2(3):148–164, 2005.

[14] C. R. Plott and S. Sunder. Efficiency of experimental security markets with insider information: An application of rational expectations models. *Journal of Political Economy*, 90:663–98, 1982.

[15] C. R. Plott and S. Sunder. Rational expectations and the aggregation of diverse information in laboratory security markets. *Econometrica*, 56:1085–1118, 1988.

[16] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach.* Prentice Hall, 2002.

[17] L. G. Valiant. The complexity of enumeration and reliability problems. *SIAM J. Comput.*, 8(3):410–421, 1979.

APPENDIX

A. APPROXIMATE PRICING OF GENERAL COMBINATORIAL MARKETS

The general problem of pricing combinatorial markets is #P-hard. In Section 3 we showed how to exactly compute asset prices for an expressive betting language for tournaments. Here we return to the general case, and present a heuristic approximation technique that, although it does not provide guaranteed bounds, we believe is practical and applicable in several settings.

Our goal as market-maker is to compute $P_q(A)$ where P_q is the probability distribution over Ω corresponding to outstanding shares q and A is an arbitrary event. Equivalently, we would like to compute $\mathbb{E}_{P_q} I_A$ where $I_A(\omega) = 1$ if $\omega \in A$ and $I_A(\omega) = 0$ otherwise. One can approximate this expectation by the unbiased estimator

$$\frac{1}{n}\sum_{i=1}^{n} I_A(X_i)$$

where $X_i \sim P_q$, i.e., X_i are draws from P_q. Since we cannot in general expect to be able to generate such draws, we rely on importance sampling [9]. The simple insight behind importance sampling is that for any measure $\mu \gg P_q$

$$\mathbb{E}_{P_q} f = \sum_{\omega\in\Omega} f(\omega)P_q(\omega) = \sum_{\omega\in\Omega} f(\omega)\frac{P_q(\omega)}{\mu(\omega)}\mu(\omega) = \mathbb{E}_\mu\left[f\frac{dP_q}{d\mu}\right].$$

Consequently, one can approximate $P_q(A)$ by the unbiased estimator

$$\frac{1}{n}\sum_{i=1}^{n} I_A(X_i)\frac{P_q(X_i)}{\mu(X_i)}$$

where $X_i \sim \mu$, i.e. X_i are draws from μ. In practice, it is useful to apply the asymptotically unbiased estimator

$$\hat{P}_q(A) = \frac{1}{\sum_{i=1}^{n} P_q(X_i)/\mu(X_i)}\sum_{i=1}^{n} I_A(X_i)\frac{P_q(X_i)}{\mu(X_i)}. \quad (3)$$

The considerable advantage of (3) is that the importance weights $P_q(X_i)/\mu(X_i)$ only need to be known up to a constant. For example, suppose we are able to draw uniformly from Ω, i.e. $\mu(\omega) = 1/N$ where $|\Omega| = N$. Then the importance weights satisfy

$$\frac{P_q(X_i)}{\mu(X_i)} = N\frac{\exp(q_{X_i}/b)}{\sum_{\omega\in\Omega}\exp(q_\omega/b)} = Z\exp(q_{X_i}/b)$$

for a constant Z. In particular, (3) simplifies to

$$\hat{P}_q(A) = \frac{1}{\sum_{i=1}^{n}\exp(q_{X_i}/b)}\sum_{X_i\in A}\exp(q_{X_i}/b). \quad (4)$$

In the above, we assumed μ to be uniform over Ω. In some cases, it may be possible to make draws from Ω according to

$$\mu(\omega) = \frac{q_\omega}{Z'}$$

where Z' is the total number of shares on Ω. Each market order $O_i = (A_i, s_i)$ consists of an event A_i and the number of shares s_i to buy on that event. Suppose that for each set corresponding to an order, we can compute its size n_i and are able to choose an outcome from A_i uniformly at random. Choose an outcome from Ω as follows:

1. Select an order O_i at random proportional to $n_i s_i$.
2. Select an outcome from O_i at random.

LEMMA A.1. *The sampling procedure above generates a draw from Ω according to the distribution $\mu(\omega) \propto q_\omega$.*

PROOF. For any outcome ω, consider the orders

$$O_{i_1}, \ldots, O_{i_m}$$

such that $\omega \in A_{i_j}$, i.e. orders where shares were purchased on ω. The number of shares on ω is then $s_{i_1} + \cdots + s_{i_m}$.

Now,

$$\begin{aligned}\mu(\omega) &= \frac{n_{i_1}s_{i_1}}{Z'}\cdot\frac{1}{n_{i_1}}+\cdots+\frac{n_{i_m}s_{i_m}}{Z'}\cdot\frac{1}{n_{i_m}}\\ &= \frac{s_{i_1}+\cdots+s_{i_m}}{Z'}\end{aligned}$$

where Z' is the total number of shares on Ω. $\square$

For $\mu(\omega)\propto q_\omega$ and $X_i\sim\mu$, we have the estimator

$$\hat{P}_q(A)=\frac{1}{\sum_{i=1}^n \exp(q_{X_i}/b)/q_{X_i}}\sum_{X_i\in A}\frac{\exp(q_{X_i}/b)}{q_{X_i}}. \tag{5}$$

B. PROOFS

LEMMA B.1. *For events $A, B\subset\Omega$, there is zero net cost for buying*

$$b\log\left(\frac{e^{\Delta/b}}{e^{\Delta/b}P_q(A|B)+P_q(\bar{A}|B)}\right)$$

shares of AB and short selling

$$b\log\left(e^{\Delta/b}P_q(A|B)+P_q(\bar{A}|B)\right)$$

shares of $\bar{A}B$.

PROOF. Letting $\tilde{q}$ denote the new distribution of shares, the cost of the transaction is

$$C(\tilde{q})-C(q)=b\log\sum_{\tau\in\Omega}e^{\tilde{q}(\tau)/b}-b\log\sum_{\tau\in\Omega}e^{q(\tau)/b}.$$

Now,

$$\begin{aligned}\frac{\sum_B e^{\tilde{q}(\tau)/b}}{\sum_B e^{q(\tau)/b}} &= \frac{\sum_{AB}e^{\tilde{q}(\tau)/b}+\sum_{\bar{A}B}e^{\tilde{q}(\tau)/b}}{\sum_B e^{q(\tau)/b}}\\ &= \frac{e^{\Delta/b}\sum_{AB}e^{q(\tau)/b}+\sum_{\bar{A}B}e^{q(\tau)/b}}{\left(e^{\Delta/b}P_q(A|B)+P_q(\bar{A}|B)\right)\sum_B e^{q(\tau)/b}}\\ &= \frac{e^{\Delta/b}P_q(AB)+P_q(\bar{A}B)}{\left(e^{\Delta/b}P_q(A|B)+P_q(\bar{A}|B)\right)P_q(B)} = 1.\end{aligned}$$

Consequently, $\sum_B e^{\tilde{q}(\tau)/b}=\sum_B e^{q(\tau)/b}$, and hence,

$$\sum_\Omega e^{\tilde{q}(\tau)/b}=\sum_\Omega e^{q(\tau)/b}.$$

$\square$

Proof of Lemma 3.4

PROOF. The cost function for the logarithmic market scoring rule is $C(q)=b\log\left(\sum_{w\in\Omega}e^{q_\omega/b}\right)$. Letting $\tilde{q}$ denote the number of shares on each state after the new shares are purchased, we have

$$\begin{aligned}C(\tilde{q})-C(q) &= b\log\left(\frac{\sum_{w\in\Omega}e^{\tilde{q}_\omega/b}}{\sum_{w\in\Omega}e^{q_\omega/b}}\right)\\ &= b\log\left(\frac{\sum_{w\in A}e^{\tilde{q}_\omega/b}+\sum_{w\notin A}e^{\tilde{q}_\omega/b}}{\sum_{w\in\Omega}e^{q_\omega/b}}\right)\\ &= b\log\left(\frac{e^{\Delta}\sum_{w\in A}e^{q_\omega/b}+\sum_{w\notin A}e^{q_\omega/b}}{\sum_{w\in\Omega}e^{q_\omega/b}}\right)\\ &= b\log\left(e^{\Delta}P(A)+P(\bar{A})\right).\end{aligned}$$

$\square$

We use the notation $A\perp_B C$ to indicate that A and C are conditionally independent given B. That is, $P(A|BC)=P(A|B)$.

LEMMA B.2. *For events A and B, suppose shares are purchased on the conditional event $A|B$. Let P denote the distribution on Ω before the shares were purchased, and let $\tilde{P}$ denote the distribution after the purchase. Then the following hold:*

1. *If $A\perp_B D$, then $\tilde{P}(D)=P(D)$*
2. *If $A\perp_{BD} C$ (or $BD=\emptyset$) and $C\perp_D B$, then $\tilde{P}(C|D)=P(C|D)$*
3. *If $A\perp_{BD} C$ (or $BD=\emptyset$) and $A\perp_B D$, then $\tilde{P}(C|D)=P(C|D)$*

where the conditional independence statements are with respect to P.

PROOF. Note that there exist c_1, c_2 such that $\tilde{P}(\omega)=c_1P(\omega)$ for $\omega\in AB$, and $\tilde{P}(\omega)=c_2P(\omega)$ for $\omega\in\bar{A}B$. Furthermore, $\tilde{P}(\omega)=P(\omega)$ for $\omega\notin B$, and $\tilde{P}(B)=P(B)$. We use the convention that, for any set S, $P(S|\emptyset)=0$. Now,

$$\begin{aligned}\tilde{P}(D) &= c_1P(ABD)+c_2P(\bar{A}BD)+P(\bar{B}D)\\ &= P(BD)[c_1P(A|BD)+c_2P(\bar{A}|BD)]+P(\bar{B}D)\\ &= P(BD)[c_1P(A|B)+c_2P(\bar{A}|B)]+P(\bar{B}D)\end{aligned}$$

where the last equality follows from the conditional independence assumption. Furthermore,

$$c_1P(A|B)+c_2P(\bar{A}|B)=\frac{c_1P(AB)+c_2P(\bar{A}B)}{P(B)}=\frac{\tilde{P}(B)}{P(B)}=1.$$

Consequently, $\tilde{P}(D)=P(D)$. To show the second statement, observe that

$$\begin{aligned}\tilde{P}(C|D) &= \frac{\tilde{P}(CD)}{\tilde{P}(D)}\\ &= \frac{\tilde{P}(ABCD)+\tilde{P}(\bar{A}BCD)+\tilde{P}(\bar{B}CD)}{\tilde{P}(D)}\\ &= \frac{c_1P(ABCD)+c_2P(\bar{A}BCD)+P(\bar{B}CD)}{\tilde{P}(D)}\\ &= \frac{c_1P(C|ABD)P(ABD)+c_2P(C|\bar{A}BD)P(\bar{A}BD)}{\tilde{P}(D)}\\ &\quad+\frac{P(C|\bar{B}D)P(\bar{B}D)}{\tilde{P}(D)}\\ &= \frac{c_1P(C|BD)P(ABD)+c_2P(C|BD)P(\bar{A}BD)}{\tilde{P}(D)}\\ &\quad+\frac{P(C|\bar{B}D)P(\bar{B}D)}{\tilde{P}(D)}\end{aligned}$$

where the last equality follows from the assumption $A\perp_{BD}$

C. Continuing this string of equalities, we have

$$\begin{aligned}\tilde{P}(C|D) &= \frac{P(C|BD)[c_1P(ABD)+c_2P(\bar{A}BD)]}{\tilde{P}(D)} \\ &\qquad + \frac{P(C|\bar{B}D)P(\bar{B}D)}{\tilde{P}(D)} \\ &= \frac{\tilde{P}(C|BD)\tilde{P}(BD)+P(C|\bar{B}D)\tilde{P}(\bar{B}D)}{\tilde{P}(D)} \\ &= P(C|BD)\tilde{P}(B|D)+P(C|\bar{B}D)\tilde{P}(\bar{B}|D). \qquad (6)\end{aligned}$$

If $C \perp_D B$, by (6) we have

$$\tilde{P}(C|D) = P(C|D)[\tilde{P}(B|D)+\tilde{P}(\bar{B}|D)] = P(C|D)$$

which proves the second statement of the theorem. For the third result, note that under the conditional independence assumption, $\tilde{P}(D) = P(D)$ by the first statement of the theorem. This implies that

$$\tilde{P}(\bar{B}|D) = \frac{\tilde{P}(\bar{B}D)}{\tilde{P}(D)} = \frac{P(\bar{B}D)}{P(D)} = P(\bar{B}|D)$$

and hence, $\tilde{P}(B|D) = P(B|D)$. Finally, from (6) we have

$$\begin{aligned}\tilde{P}(C|D) &= P(C|BD)P(B|D)+P(C|\bar{B}D)P(\bar{B}|D) \\ &= P(CB|D)+P(C\bar{B}|D) \\ &= P(C|D).\end{aligned}$$

□

Proof of Theorem 3.4

PROOF. Each interior (i.e., non-leaf) node X_i has exactly two parents, which we denote by $\hat{X}_i^L$ and $\hat{X}_i^R$. For the leaf nodes, we use the convention that $\hat{X}_i^L$ and $\hat{X}_i^R$ are the two teams which (deterministically) play in the game corresponding to X_i. Now, for the uniform distribution P_0, and $i > 1$

$$P_0(X_i = x_i|X_1,\ldots,X_{i-1}) = \begin{cases} 1/2 & \hat{X}_i^L = x_i \text{ or } \hat{X}_i^R = x_i \\ 0 & \text{otherwise} \end{cases}.$$

In particular,

$$P_0(X_i = x_i|X_1,\ldots,X_{i-1}) = P_0(X_i = x_i|X_i^L, X_i^R)$$

and so P_0 is represented by the Bayesian network.

For $i > 1$, set $A = \{X_{g_j} = t_j\}$, $B = \{\{\hat{X}_{g_j}^L, \hat{X}_{g_j}^R\} = \{t_j, t_j'\}\}$, $C = \{X_i = x_i\}$, $D = \{X_1 = x_1, \ldots, X_{i-1} = x_{i-1}\}$ for some configuration $(x_1,\ldots,x_{i-1})$, and $\hat{D} = \{\hat{X}_i^L = \hat{x}_i^L, \hat{X}_i^R = \hat{x}_i^R\}$ where $\hat{x}_i^L$ and $\hat{x}_i^R$ are the values assigned in D. Consider the following cases:

1. $g_j < i$: If $BD \neq \emptyset$, then $P_j(A|CBD) = P_j(A|BD)$. So $A \perp_{BD} C$, or $BD = \emptyset$. Also, $P_j(B|CD) = P_j(B|D)$, so $C \perp_D B$. Consequently, by Lemma B.2(2), $\tilde{P}(C|D) = P(C|D) = P(C|\hat{D})$.
2. $g_j > i$: If $BD \neq \emptyset$, then $P(A|CBD) = P(A|B) = P(A|BD)$. So $A \perp_{BD} C$, or $BD = \emptyset$. Also, $P(A|BD) = P(A|B)$, so $A \perp_B D$. Consequently, by Lemma B.2(3), $\tilde{P}(C|D) = P(C|D) = P(C|\hat{D})$.
3. $g_j = i$: In this case, either $D \subset B$ or $D \subset \bar{B}$. If $D \subset \bar{B}$, then

$$\tilde{P}(C|D) = \frac{\tilde{P}(CD)}{\tilde{P}(D)} = \frac{P(CD)}{P(D)} = P(C|D) = P(C|\hat{D}).$$

Now we consider $D \subset B$. Then,

$$\begin{aligned}\tilde{P}(C|D) &= \frac{\tilde{P}(CD)}{\tilde{P}(D)} = \frac{c_1P(CDA)+c_2P(CD\bar{A})}{c_1P(DA)+c_2P(D\bar{A})} \\ &= \frac{c_1P(CA|D)+c_2P(C\bar{A}|D)}{c_1P(A|D)+c_2P(\bar{A}|D)} \\ &= \frac{c_1P(CA|\hat{D})+c_2P(C\bar{A}|\hat{D})}{c_1P(A|\hat{D})+c_2P(\bar{A}|\hat{D})}\end{aligned}$$

Since $D \subset B$, $\hat{D} = \hat{B}$ and the denominator above equals 1. Consequently,

$$\tilde{P}(C|D) = c_1P(CA|\hat{D}) + c_2P(C\bar{A}|\hat{D}).$$

In all three cases, we have that $\tilde{P}(C|D)$ depends only on $\hat{D}$, and furthermore, only the distribution of game i changes. □

Proof of Corollary 3.6

PROOF. The result follows from the construction given in the proof of Theorem 3.4, with values of c_1 and c_2 derived from (1) and (2). □