A Tractable Combinatorial Market Maker Using Constraint Generation

MIROSLAV DUDÍK, Yahoo! Research
SEBASTIEN LAHAIE, Yahoo! Research
DAVID M. PENNOCK, Yahoo! Research

We present a new automated market maker for providing liquidity across multiple logically interrelated securities. Our approach lies somewhere between the industry standard—treating related securities as independent and thus not transmitting any information from one security to another—and a full combinatorial market maker for which pricing is computationally intractable. Our market maker, based on convex optimization and constraint generation, is tractable like independent securities yet propagates some information among related securities like a combinatorial market maker, resulting in more complete information aggregation. We prove several favorable properties of our scheme and evaluate its information aggregation performance on survey data involving hundreds of thousands of complex predictions about the 2008 U.S. presidential election.

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

General Terms: Algorithms, Economics

Additional Key Words and Phrases: Prediction market, market maker, combinatorial security, independent markets, convex optimization

1. INTRODUCTION

During the Republican presidential primary in 2011–2012, the Irish prediction market Intrade featured a number of interrelated propositions, including "Mitt Romney to win Iowa", "Romney to finish second in Iowa", and "Romney to sweep the first five primaries including Iowa". As is standard practice from Wall Street to Las Vegas, Intrade operated these securities independently. For example, as far as Intrade was concerned, the price of "Romney to sweep" could eclipse the price of "Romney to win Iowa", representing a logical impossibility.

A combinatorial security is a financial instrument whose payoff is an arbitrary function of a common set of input variables. (In our examples, we consider Boolean functions of binary literals.) When clear from context, we sometimes refer to a security and its payoff function interchangeably.

It is easy to see that enforcing all constraints among combinatorial securities is intractable. Consider two Boolean securities A and B whose prices sum to more than 1. If they represent disjoint events, their probabilities must sum to 1 or less, which presents an arbitrage opportunity. However, determining whether the two securities are disjoint is equivalent to determining whether $A \cap B$ is unsatisfiable, an NP-hard problem. Note that, in general, finding arbitrage in a system of combinatorial securities is NP-hard even if all securities are conjunctions of only two (positive or negative) literals [Fortnow et al. 2004]. Maintaining point prices consistent with a standard

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
EC'12, June 4–8, 2012, Valencia, Spain. Copyright 2012 ACM 978-1-4503-1415-2/12/06...$10.00.

market maker function such as Hanson's [2003; 2007] logarithmic market scoring rule (LMSR) is #P-hard, or as hard as satisfiability counting [Chen et al. 2008a].

We develop a method for propagating as much information as possible among logically related securities while keeping our pricing algorithm tractable. By relieving traders from correcting inconsistencies (a task that computers are far better at anyway), we free them to concentrate on providing information in whatever form they find natural. To the extent that the goal of a prediction market is to aggregate information—and thus to reward traders for providing actual information and not for their computing horsepower—automating the mechanical task of logical inference should be beneficial.

As a running example, we consider the 2012 U.S. presidential election between Democrat Barack Obama and Republican Mitt Romney.[1] Votes in each of the fifty U.S. states and the District of Columbia are tallied separately, and each state has its own election result—either a win for the Republicans or the Democrats. The final national winner is the candidate who earns a weighted majority of all the individual states, where a state's weight (number of electoral votes) roughly corresponds to its population.[2]

In the election market, traders can buy base securities like "Republicans to win Florida" or "Democrats to win Pennsylvania", or derived securities such as

— Democrats to win PA and OH, and Republicans to win FL
— The same party to win both OH and FL
— Democrats to win 9 of the 10 northeastern states, or
— Republicans to win the national election.

Note that, even restricting to conjunctions, the number of possible events is 3^{51}, exponentially large. Yet, we would like to allow interested traders to buy and sell any security from this set.

Intrade is an exchange: it only matches up willing traders and takes no positions or risk of its own. Operating a *combinatorial exchange*, beyond the daunting computational challenge [Fortnow et al. 2004], is impractical for a more pedestrian reason: it is hard to imagine how any given trade, selected from the unimaginably large sea of choices, would happen to have a matching counter trade waiting to execute against it. For this reason, an automated market maker, which always offers a price for any security no matter how complex, is better suited for combinatorial prediction markets.

Unlike an exchange, a market maker by definition exposes itself to risk. Still, many market makers have a provable worst-case upper bound on loss that's logarithmic in—or even independent of—the number of outcomes and holds regardless of the true outcome or the sequence of trades [Chen and Pennock 2007; Othman and Sandholm 2011]. An exact combinatorial market maker fully propagates all information across all securities, explicitly or implicitly maintaining a complete and consistent joint probability distribution over the exponentially large outcome space. As a result, prices cannot yield arbitrage opportunities.

Operating an exact combinatorial market maker is computationally even harder than running an exchange [Chen et al. 2008a]. One way to recover efficiency is to restrict the set of tradable securities [Abernethy et al. 2011; Agrawal et al. 2008; Chen et al. 2007, 2008b; Guo and Pennock 2009; Pennock and Xia 2011]. Most of these results are negative, with even mild forms of expressiveness remaining intractable. A

[1] For simplicity of exposition we assume no additional candidates, but our formalism can accommodate three or more candidates.

[2] This is again a simplification: Maine and Nebraska may split their electoral votes, but we can handle this by splitting these states into multiple sub-states.

second solution, and the one favored in practice, is to simply ignore the relationships among securities and create independent prediction markets for each security or small groups of securities for which prices can be efficiently calculated. This approach is simple and fully general, yet does nothing to limit arbitrage and exposes the market maker to the risk of exponentially large losses.

In this paper we propose a solution that lies between these two approaches. We implement the pricing as in the independent-markets solution, but we detect and eliminate many forms of arbitrage using optimization and constraint generation. As a result, our market maker is tractable, propagates a large amount of information, and maintains a reasonable loss bound.

Our method is inspired by Abernethy et al. [2011], who give a broad characterization of arbitrage-free and bounded-loss pricing rules. Their pricing rules are derived from smooth and convex *cost functions* defined over share vectors corresponding to quantities of individual securities sold by the market maker. Prices correspond to partial derivatives of the cost. Abernethy et al. [2011] show that prices generated by the cost-based mechanism yield bounded loss and are arbitrage-free as long as the set of price vectors (across all possible allocations) coincides with the convex hull of payoff vectors (across all possible outcomes), which we call the *realizable polytope*.

The independent-markets solution generates price vectors that lie in a superset of the realizable polytope. Our arbitrage detection can be viewed as an incomplete separation oracle for the realizable polytope. Rather than detecting separating hyperplanes between an arbitrary point and the realizable polytope, we only provide a separation oracle for a relaxed version of the realizable polytope. Thus, even after our detectors exhaust all of the arbitrage opportunities they can find, there may be some residual arbitrage due to the relaxation of the realizable polytope. The problem of approximating the realizable polytope is an active area of research in machine learning that arises in the context of variational inference methods for graphical models [Wainwright and Jordan 2008; Sontag and Jaakkola 2007], and these connections underlie our approach to constraint-based market making.

Abernethy et al. [2011] also propose a mechanism based on a separation oracle, but they require that it be repeatedly called in each pricing computation until no more arbitrage is detected. This is impractical for even moderately sized prediction markets since pricing computations need to be efficient. Here, we decouple pricing computations (which are done efficiently as in independent markets) from arbitrage detection, which is key for practical implementations. This modification requires a new analysis of the market maker's loss. The second generalization that we make is dropping Abernethy et al.'s assumption of the "realizable market initialization".

Apart from the theoretical contribution, we perform an extensive evaluation of our mechanism on over 300,000 probability assessments from the 2008 U.S. presidential elections collected in a Princeton University survey [Wang et al. 2011]. We assess the forecasting performance of our approach against competitive benchmarks and study the impact of various separation-oracle choices. Apart from small-scale laboratory studies [Hanson et al. 2007], we are not aware of previous empirical evaluations of combinatorial prediction markets, so we believe this is a significant contribution in its own right and, to our knowledge, represents the largest-scale evaluation of a prediction market to date.

2. PRELIMINARIES

2.1. Logarithmic market scoring rule

We begin with a simple example of a market maker based on the *logarithmic market scoring rule* (LMSR) [Hanson 2003, 2007]. Let Ω be a finite set of *outcomes*, which

correspond to mutually exclusive and exhaustive states of the world. Each outcome $\omega \in \Omega$ is associated with an *atomic security*, which pays out \$1 if the outcome occurs and \$0 otherwise. The state of the market is specified by a vector $\boldsymbol{\theta} \in \mathbb{R}^\Omega$ listing the number of shares of each atomic security sold by the market maker. Security prices are defined using the cost function

$$C(\boldsymbol{\theta}) = B \ln \left(\sum_{\omega \in \Omega} e^{\theta_\omega / B} \right) \ ,$$

where B is the liquidity parameter controlling the rate at which prices change. We consider traders buying *bundles* $\boldsymbol{\delta} \in \mathbb{R}^\Omega$ of atomic securities (negative entries correspond to short-selling). A trader holding a bundle $\boldsymbol{\delta}$ receives the payoff δ_ω if outcome ω is realized. A trader wishing to buy a bundle $\boldsymbol{\delta}$ in the market state $\boldsymbol{\theta}$ must pay $C(\boldsymbol{\theta}+\boldsymbol{\delta}) - C(\boldsymbol{\theta})$ to the market maker, after which the new state becomes $\boldsymbol{\theta}+\boldsymbol{\delta}$. Thus, the instantaneous price of security ω is

$$p_\omega(\boldsymbol{\theta}) = \partial C(\boldsymbol{\theta}) / \partial \theta_\omega = \frac{e^{\theta_\omega / B}}{\sum_{\omega' \in \Omega} e^{\theta_{\omega'} / B}} \ .$$

The cost of a bundle $\boldsymbol{\delta} \in \mathbb{R}^\Omega$ can be expressed using the instantaneous price vector as:

$$C(\boldsymbol{\theta} + \boldsymbol{\delta}) - C(\boldsymbol{\theta}) = B \ln \left(\sum_{\omega \in \Omega} p_\omega(\boldsymbol{\theta}) e^{\delta_\omega / B} \right) \ . \tag{1}$$

Note that the vector $\boldsymbol{p}(\boldsymbol{\theta})$ is a probability distribution over Ω and can thus be interpreted as the market's current belief about the state of the world. Moreover, the set of price vectors $\boldsymbol{p}(\boldsymbol{\theta})$ across all $\boldsymbol{\theta} \in \mathbb{R}^\Omega$ corresponds to the interior of the $|\Omega|$-dimensional simplex. Thus, any belief that assigns non-zero probability to every outcome is expressible, given sufficient purchasing power of the traders.

For an arbitrary subset $E \subseteq \Omega$, called an *event*, we can define an Arrow-Debreu security which pays out \$1 if and only if $\omega \in E$. Such a security can be modeled as a bundle $\boldsymbol{\delta}$ with $\delta_\omega = \mathbf{1}[\omega \in E]$, where we use the notation $\mathbf{1}[\cdot]$ for an indicator equal to 1 if the enclosed statement is true and 0 otherwise. The instantaneous price of such a bundle corresponds to the probability of the event E under the market's belief.

In the election market example, the set of outcomes is $\Omega = \{0,1\}^{51}$ (where 0 indicates Democrats, and 1 indicates Republicans winning a given state). The event "Democrats to win OH" then corresponds to the set E consisting of outcomes ω that assign 0 to OH. Calculating LMSR prices according to the formula above requires summation over Ω, and is therefore intractable. This intractability appears unavoidable since exact LMSR pricing of even pairwise conjunctions is #P-hard [Chen et al. 2008a]. On the other hand, Xia and Pennock [2011] and Chen et al. [2008b] discuss approximating LMSR prices using importance sampling, Metropolis-Hastings, or Monte Carlo methods. For example, Predictalot, a combinatorial prediction market game built and operated by Yahoo Labs for NCAA men's college basketball in 2010 and 2011, Soccer World Cup 2010, and India Cricket World Cup 2011, employed a naive version of importance sampling and attracted thousands of fans. Still, the volatility of approximate prices reduces the user experience and makes controlling market maker loss and allowing limit orders more difficult.

2.2. Cost-based market making

We introduce a generalization of LMSR due to Abernethy et al. [2011]. Results in this section and the next are largely taken from Abernethy et al. [2011], with the exception of Thm. 2.2, which is new. We use a slightly different notation and focus primarily on the cost function rather than its conjugate.

As before, let Ω be the set of outcomes, which we no longer require to be finite. We consider a set of securities indexed by $i \in \mathcal{I}$ where $\mathcal{I}$ is a finite index set. Each security

is associated with a *payoff function* $\phi_i : \Omega \to \mathbb{R}$, with $\phi_i(\omega)$ corresponding to the payoff for each share of security i if the outcome ω occurs. If a trader holds a bundle $\boldsymbol{\delta} \in \mathbb{R}^{\mathcal{I}}$, and the outcome ω occurs, she receives a payoff of $\boldsymbol{\delta} \cdot \boldsymbol{\phi}(\omega)$. Our previous example is recovered by setting $\mathcal{I} = \Omega$ and $\phi_i(\omega) = \mathbf{1}[i = \omega]$.

The state of the market is determined by an allocation vector $\boldsymbol{\theta} \in \mathbb{R}^{\mathcal{I}}$, and security prices are determined by a differentiable convex cost function $C : \mathbb{R}^{\mathcal{I}} \to \mathbb{R}$. As in LMSR, to buy a bundle $\boldsymbol{\delta}$ in a market state $\boldsymbol{\theta}$, a trader is charged $C(\boldsymbol{\theta} + \boldsymbol{\delta}) - C(\boldsymbol{\theta})$. Thus, the *cost-based market maker* is fully specified by a tuple $(\Omega, \mathcal{I}, \boldsymbol{\phi}, C)$.

The gradient of the cost function corresponds to the vector of instantaneous prices: $\boldsymbol{p}(\boldsymbol{\theta}) = \nabla C(\boldsymbol{\theta})$. Thinking about $\boldsymbol{p}$ as a map $\boldsymbol{p} : \mathbb{R}^{\mathcal{I}} \to \mathbb{R}^{\mathcal{I}}$, we write $\operatorname{im} \boldsymbol{p}$ for the image of $\boldsymbol{p}$, namely the set of all possible price vectors induced by C. While in LMSR instantaneous prices model probabilities, in this setting, prices model expected payoffs. If a risk-neutral trader believes that the expected payoff for a bundle $\boldsymbol{\delta}$ is above $\boldsymbol{\delta} \cdot \boldsymbol{p}(\boldsymbol{\theta})$, she is incentivized to buy the bundle. In LMSR, we were able to express all possible (non-degenerate) beliefs. Here, we are also interested in cost functions that allow expressing all realizable expectations. Let

$$\mathcal{M} = \operatorname{co}\{\boldsymbol{\phi}(\omega) : \omega \in \Omega\}$$

denote the convex hull of the payoff vectors across all outcomes. Note that points in $\mathcal{M}$ are exactly all realizable vectors of expected payoffs. We refer to $\mathcal{M}$ as the *realizable polytope*.

Let cl denote the closure of a set. The following results, proved by Abernethy et al. [2011], show that the correspondence between $\operatorname{cl}(\operatorname{im} \boldsymbol{p})$ and $\mathcal{M}$ that we observed for LMSR is not just a coincidence, but a necessary condition for arbitrage-free bounded-loss market making:

THEOREM 2.1 (SEE THMS. 2 AND 5 OF ABERNETHY ET AL. [2011]). *For any cost-based market maker:*

(*1*) *The worst-case loss is unbounded if* $\mathcal{M} \not\subseteq \operatorname{cl}(\operatorname{im} \boldsymbol{p})$.
(*2*) *Prices are arbitrage-free if and only if* $\operatorname{im} \boldsymbol{p} \subseteq \mathcal{M}$.

2.3. Market maker's loss: Convex conjugacy and Bregman divergence

To quantify the loss of a market maker, we next introduce the convex conjugate of the cost function. Compared with Thm. 6 of Abernethy et al. [2011], our bound (Thm. 2.2) makes fewer assumptions, and in particular, we do not require that the market starts in a state $\boldsymbol{\theta}^0$ such that $\boldsymbol{p}(\boldsymbol{\theta}^0) \in \mathcal{M}$ (i.e., we do not require a "realizable initialization").

Given a market maker with the cost function C, the *convex conjugate* of C is the function $R : \mathbb{R}^{\mathcal{I}} \to (-\infty, \infty]$ defined by

$$R(\boldsymbol{\mu}) = \sup_{\boldsymbol{\theta} \in \mathbb{R}^{\mathcal{I}}} [\boldsymbol{\theta} \cdot \boldsymbol{\mu} - C(\boldsymbol{\theta})] \quad . \tag{2}$$

From convex analysis [Rockafellar 1970, Thm. 12.2 and Cor. 26.4.1], R is a convex function, with $R(\boldsymbol{\mu}) = \infty$ for $\boldsymbol{\mu} \notin \operatorname{cl}(\operatorname{im} \boldsymbol{p})$. By checking the first order optimality conditions in Eq. (2), it is also easy to see that for $\boldsymbol{\mu} = \boldsymbol{p}(\boldsymbol{\theta}) = \nabla C(\boldsymbol{\theta})$, we have

$$R(\boldsymbol{p}(\boldsymbol{\theta})) = \boldsymbol{\theta} \cdot \boldsymbol{p}(\boldsymbol{\theta}) - C(\boldsymbol{\theta}) \quad . \tag{3}$$

For $\boldsymbol{\mu} \in \operatorname{cl}(\operatorname{im} \boldsymbol{p}) \setminus (\operatorname{im} \boldsymbol{p})$ the value can be either finite or infinite (depending on C). For LMSR, $\operatorname{cl}(\operatorname{im} \boldsymbol{p})$ is the simplex, and R is equal to negative entropy:

$$R(\boldsymbol{\mu}) = \begin{cases} \sum_{\omega \in \Omega} \mu_\omega \ln \mu_\omega & \text{if } \boldsymbol{\mu} \text{ is a probability distribution on } \Omega \\ \infty & \text{otherwise.} \end{cases}$$

Since, by our definition, C is continuous and convex, from convex analysis [Rockafellar 1970, Cor. 12.2.1] we also have the double conjugacy relationship:

$$C(\boldsymbol{\theta}) = \sup_{\boldsymbol{\mu} \in \mathbb{R}^{\mathcal{I}}} \left[\boldsymbol{\theta} \cdot \boldsymbol{\mu} - R(\boldsymbol{\mu}) \right] \ . \tag{4}$$

Comparing Eq. (3) with Eq. (4), we obtain that the supremum of Eq. (4) is attained at $\boldsymbol{\mu}^* = \boldsymbol{p}(\boldsymbol{\theta})$; it turns out that this is the unique $\boldsymbol{\mu}^*$ attaining the supremum [Rockafellar 1970, Thm. 23.5.a,b*].

For $\boldsymbol{\theta} \in \mathbb{R}^{\mathcal{I}}$ and $\boldsymbol{\mu} \in \mathbb{R}^{\mathcal{I}}$, the *mixed Bregman divergence* is defined as

$$D(\boldsymbol{\mu} \| \boldsymbol{\theta}) = R(\boldsymbol{\mu}) + C(\boldsymbol{\theta}) - \boldsymbol{\theta} \cdot \boldsymbol{\mu} \ .$$

By Eq. (4), Bregman divergence is always non-negative. Since Eq. (4) is uniquely maximized by $\boldsymbol{\mu}^* = \boldsymbol{p}(\boldsymbol{\theta})$, we also obtain that $D(\boldsymbol{\mu} \| \boldsymbol{\theta}) = 0$ if and only if $\boldsymbol{\mu} = \boldsymbol{p}(\boldsymbol{\theta})$. Hence, Bregman divergence can be viewed as a measure of distance between $\boldsymbol{\mu}$ and $\boldsymbol{p}(\boldsymbol{\theta})$. For LMSR, it corresponds to the Kullback-Leibler divergence:

$$D(\boldsymbol{\mu} \| \boldsymbol{\theta}) = \begin{cases} \sum_{\omega \in \Omega} \mu_\omega \ln \left(\mu_\omega / p_\omega(\boldsymbol{\theta}) \right) & \text{if } \boldsymbol{\mu} \text{ is a probability distribution on } \Omega \\ \infty & \text{otherwise.} \end{cases}$$

Bregman divergence bounds the loss of a market maker:

THEOREM 2.2. *Let $\boldsymbol{\theta}^0$ be the initial state of the market. If outcome ω occurs, the loss of the market maker is no more than $D(\boldsymbol{\phi}(\omega) \| \boldsymbol{\theta}^0)$.*

PROOF. Let $\boldsymbol{\theta}$ be the final state of the market. The loss of the market maker is then

$$\begin{aligned} \mathit{loss}(\omega) &= (\boldsymbol{\theta} - \boldsymbol{\theta}^0) \cdot \boldsymbol{\phi}(\omega) - C(\boldsymbol{\theta}) + C(\boldsymbol{\theta}^0) \\ &= \left[\boldsymbol{\theta} \cdot \boldsymbol{\phi}(\omega) - C(\boldsymbol{\theta}) \right] - \boldsymbol{\theta}^0 \cdot \boldsymbol{\phi}(\omega) + C(\boldsymbol{\theta}^0) \\ &\leq R(\boldsymbol{\phi}(\omega)) - \boldsymbol{\theta}^0 \cdot \boldsymbol{\phi}(\omega) + C(\boldsymbol{\theta}^0) \\ &= D(\boldsymbol{\phi}(\omega) \| \boldsymbol{\theta}^0) \end{aligned}$$

where the middle inequality follows by Eq. (2). □

The key insight here is that for bounded loss we do not need to have the correspondence $\mathrm{cl}(\mathrm{im}\, \boldsymbol{p}) = \mathcal{M}$, and we do not even need to have realizable initialization $\boldsymbol{p}(\boldsymbol{\theta}^0) \in \mathcal{M}$. All that is required is that $D(\boldsymbol{\mu} \| \boldsymbol{\theta}^0)$ be bounded for all $\boldsymbol{\mu} \in \mathcal{M}$. For example, for LMSR,

$$\sup_{\boldsymbol{\mu} \in \mathcal{M}} D(\boldsymbol{\mu} \| \boldsymbol{\theta}^0) = \max_{\omega \in \Omega} \ln(1 / p_\omega(\boldsymbol{\theta}^0)) \ ,$$

which is finite.

Perhaps not surprisingly, Bregman divergence D quantifies not only the loss of the market maker, but also the gain of an arbitrager:

THEOREM 2.3 (THM. 7 OF ABERNETHY ET AL. 2011). *If the market is in a state $\boldsymbol{\theta}$, a trader has an opportunity to earn a guaranteed profit of at least $\min_{\boldsymbol{\mu} \in \mathcal{M}} D(\boldsymbol{\mu} \| \boldsymbol{\theta})$.*

3. MARKET MAKING WITH LINEAR CONSTRAINTS

Our design has two components. First, we break a large market into small markets, which solves the intractability, but may introduce arbitrage. Second, we introduce a general scheme for detecting and removing arbitrage based on a formalization of realizability using linear equalities and inequalities. In this section, we discuss these two components in more detail and also introduce two specific algorithms for arbitrage removal (projected gradient and coordinate descent). In the next section we provide specific algorithms for arbitrage detection in the elections market.

3.1. Independent markets

One general approach to tractable market making with bounded loss is to partition $\mathcal{I}$ into groups $g \subseteq \mathcal{I}$ and use tractable bounded-loss cost functions C_g for each group g, effectively treating each group as a separate market.

Formally, let $\mathcal{G}$ be a system of non-empty sets $g \subseteq \mathcal{I}$ such that $\mathcal{I} = \biguplus_{g \in \mathcal{G}} g$, where $\biguplus$ denotes disjoint union. The block of coordinates θ_i for $i \in g$ is written as $\boldsymbol{\theta}_g$. Let $C_g : \mathbb{R}^g \to \mathbb{R}$ for $g \in \mathcal{G}$ be differentiable convex functions. Define

$$C(\boldsymbol{\theta}) = \sum_{g \in \mathcal{G}} C_g(\boldsymbol{\theta}_g) \ .$$

From this, we can easily derive that $\boldsymbol{p}(\boldsymbol{\theta}) = (\boldsymbol{p}_g(\boldsymbol{\theta}_g))_{g \in \mathcal{G}}$. We assume that the computation of C_g and $\boldsymbol{p}_g = \nabla C_g$ is tractable, hence so is the computation of C and $\boldsymbol{p}$. If the market is initialized at a state $\boldsymbol{\theta}^0$ so that $D_g(\boldsymbol{\mu}_g \| \boldsymbol{\theta}^0_g)$ is bounded, we also have that $D(\boldsymbol{\mu} \| \boldsymbol{\theta}^0) = \sum_{g \in \mathcal{G}} D_g(\boldsymbol{\mu}_g \| \boldsymbol{\theta}^0_g)$ is bounded. However, this bound will typically be worse than for an LMSR-style market, especially if there is a large degree of deterministic dependence among payoffs in individual groups.

In the election market, we consider securities associated with base events, such as "Democrats to win FL", as well as securities associated with conjunctions and disjunctions of base events. Base-event securities appear in groups of two, such as {"Democrats to win FL", "Republicans to win FL"}. Pairwise conjunctions appear in groups of four, corresponding to disjoint exhaustive events of the form $\{A \cap B,\ A \cap B^c,\ A^c \cap B,\ A^c \cap B^c\}$, where A and B are base events. Conjunctions and disjunctions of three or more base events appear in groups of two, such as $\{A \cap B \cap C,\ A^c \cup B^c \cup C^c\}$.

If in the election market example we use LMSR with the same liquidity parameter for each group, the worst-case loss bound, while finite, will be linear in the number of groups, which is exponential in the number of base events. In contrast, LMSR on the entire outcome space suffers a loss which is at most logarithmic in the outcome-set cardinality, or linear in the number of base events.

There are three avenues for improving the independent-markets loss bound. First, even though there is an exponential number of securities (and groups of securities), not all of them will be traded. Groups that are never traded can be ignored in cost calculations and do not contribute to the loss bound. In particular, during the run of a mechanism, the number of traded groups grows at most linearly with the number of realized trades. Second, each group can in principle have its own liquidity parameter B_g, which functions as a loss-bound coefficient for that group. It is possible to choose liquidity parameters that reflect the "complexity" of a security (smaller B_g for groups with more complex securities) and obtain a better upper bound on the final loss. For example, positing a probability measure $\beta(i)$ over securities, we can set $B_g = B\beta(g)$ and guarantee the loss bound proportional to B regardless of the number of groups (assuming sensible initialization for each group). This allows treatment of not only finite but also countable index sets $\mathcal{I}$. The third approach, which is the subject of the next section, is based on the observation that there are many easy-to-detect arbitrage opportunities in the independent markets, and the market maker can recoup some of its losses by acting as an arbitrager. Apart from eliminating potential losses, this empirically results in better information aggregation, as we will see in Sec. 5.

3.2. Arbitrage detection and minimization

From Thm. 2.3, it is clear that ensuring realizable prices helps eliminate arbitrage opportunities. Given the hardness results for even just pairwise conjunctions, we do not attempt to efficiently describe the realizable polytope $\mathcal{M}$, and instead will only

assume access to an efficient representation of a superset $\tilde{\mathcal{M}} \supseteq \mathcal{M}$. In particular, we assume that $\tilde{\mathcal{M}}$ is described by linear inequalities

$$\tilde{\mathcal{M}} = \{\boldsymbol{\mu} \in \mathbb{R}^{\mathcal{I}} : \mathbf{A}^\top \boldsymbol{\mu} \geq \mathbf{b}\}$$

where $\mathbf{A} \in \mathbb{R}^{\mathcal{I} \times \mathcal{K}}$ and $\mathbf{b} \in \mathbb{R}^{\mathcal{K}}$. Columns of $\mathbf{A}$ are denoted $\mathbf{a}_k$ where $k \in \mathcal{K}$ is the index over linear constraints. Thus, realizability constraints are of the form:

$$\mathbf{a}_k \cdot \boldsymbol{\mu} \geq b_k \ .$$

Equalities can also be represented as pairs of inequalities. As a special case, $\boldsymbol{\phi}(\omega)$ must satisfy these constraints regardless of the outcome ω. This means that the vector $\mathbf{a}_k \in \mathbb{R}^{\mathcal{I}}$ can be viewed as a bundle with the guaranteed payoff of at least b_k. Thus, whenever the price of $\mathbf{a}_k$ drops below b_k, i.e.,

$$\mathbf{a}_k \cdot \boldsymbol{p}(\boldsymbol{\theta}) < b_k \ ,$$

there is an arbitrage opportunity. In such situations, we allow the market maker to buy this bundle and move the market away from arbitrage.

We extend the market maker's state by a non-negative vector $\boldsymbol{\eta} \in \mathbb{R}_+^{\mathcal{K}}$, whose entries correspond to the amount of each of the arbitrage bundles that the market maker bought; thus, $\boldsymbol{\eta}$ represents a "bundle of bundles", which translates to the bundle $\mathbf{A}\boldsymbol{\eta}$ in the original security space. This leads to the definition of the *extended cost* of the market maker:

$$\tilde{C}(\boldsymbol{\theta}, \boldsymbol{\eta}) = C(\boldsymbol{\theta} + \mathbf{A}\boldsymbol{\eta}) - \mathbf{b} \cdot \boldsymbol{\eta}$$

where $\boldsymbol{\theta} \in \mathbb{R}^{\mathcal{I}}$ indicates amounts of shares bought by traders and $\boldsymbol{\eta} \in \mathbb{R}_+^{\mathcal{K}}$ is an internal vector of the market maker. The addition of $\mathbf{A}\boldsymbol{\eta}$ accounts for the purchase of $\mathbf{A}\boldsymbol{\eta}$ original securities by the market maker, subtraction of $\mathbf{b} \cdot \boldsymbol{\eta}$ accounts for the guaranteed payoff to self. The market maker follows the following protocol:

ALGORITHM 1: Cost-based Market Making with Linear Constraints

Input:

outcome space Ω, security space $\mathcal{I}$, payoff function $\boldsymbol{\phi}$, cost function C, initial state $\boldsymbol{\theta}^0$
matrix $\mathbf{A}$ and vector $\mathbf{b}$ defining $\tilde{\mathcal{M}} \supseteq \mathcal{M}$

Protocol:

initialize $\boldsymbol{\eta}^0 = \mathbf{0}$
for $t = 1, \ldots, T$ (where T is an a priori unknown number of trades):
  receive a request for a bundle $\boldsymbol{\delta}^t$
  sell the bundle $\boldsymbol{\delta}^t$ for the cost $\tilde{C}(\boldsymbol{\theta}^{t-1} + \boldsymbol{\delta}^t, \boldsymbol{\eta}^{t-1}) - \tilde{C}(\boldsymbol{\theta}^{t-1}, \boldsymbol{\eta}^{t-1})$
  let $\boldsymbol{\theta}^t = \boldsymbol{\theta}^{t-1} + \boldsymbol{\delta}^t$
  choose $\boldsymbol{\eta}^t \in \mathbb{R}_+^{\mathcal{K}}$ such that $\tilde{C}(\boldsymbol{\theta}^t, \boldsymbol{\eta}^t) \leq \tilde{C}(\boldsymbol{\theta}^t, \boldsymbol{\eta}^{t-1})$
observe ω
pay $(\boldsymbol{\theta}^T - \boldsymbol{\theta}^0) \cdot \boldsymbol{\phi}(\omega)$ to traders

Let $\Delta^t = \tilde{C}(\boldsymbol{\theta}^t, \boldsymbol{\eta}^{t-1}) - \tilde{C}(\boldsymbol{\theta}^t, \boldsymbol{\eta}^t) \geq 0$ denote the improvement in the extended cost due to updating $\boldsymbol{\eta}$ after the t-th transaction. Next we prove that the extended-cost market maker has bounded loss, provided that the initial cost function C yielded bounded loss and that updating $\boldsymbol{\eta}$ yields better loss bounds.

THEOREM 3.1. *Let $\boldsymbol{\theta}^0$ be the initial market state. If outcome ω occurs, the loss of the cost-based market maker with linear constraints is at most $D(\boldsymbol{\phi}(\omega) \| \boldsymbol{\theta}^0) - \sum_{t=1}^{T} \Delta^t$.*

PROOF. According to the protocol above, the loss of the market maker is

$$\begin{aligned}
loss(\omega) &= (\boldsymbol{\theta}^T - \boldsymbol{\theta}^0) \cdot \boldsymbol{\phi}(\omega) - \textstyle\sum_{t=1}^T \left[\tilde{C}(\boldsymbol{\theta}^t, \boldsymbol{\eta}^{t-1}) - \tilde{C}(\boldsymbol{\theta}^{t-1}, \boldsymbol{\eta}^{t-1})\right] \\
&= (\boldsymbol{\theta}^T - \boldsymbol{\theta}^0) \cdot \boldsymbol{\phi}(\omega) - \textstyle\sum_{t=1}^T \left[\Delta^t + \tilde{C}(\boldsymbol{\theta}^t, \boldsymbol{\eta}^t) - \tilde{C}(\boldsymbol{\theta}^{t-1}, \boldsymbol{\eta}^{t-1})\right] \\
&= (\boldsymbol{\theta}^T - \boldsymbol{\theta}^0) \cdot \boldsymbol{\phi}(\omega) - \tilde{C}(\boldsymbol{\theta}^T, \boldsymbol{\eta}^T) + \tilde{C}(\boldsymbol{\theta}^0, \boldsymbol{\eta}^0) - \textstyle\sum_{t=1}^T \Delta^t \\
&= (\boldsymbol{\theta}^T - \boldsymbol{\theta}^0) \cdot \boldsymbol{\phi}(\omega) - C(\boldsymbol{\theta}^T + \mathbf{A}\boldsymbol{\eta}^T) + \mathbf{b} \cdot \boldsymbol{\eta}^T + C(\boldsymbol{\theta}^0) - \textstyle\sum_{t=1}^T \Delta^t \\
&= \left[(\boldsymbol{\theta}^T + \mathbf{A}\boldsymbol{\eta}^T) \cdot \boldsymbol{\phi}(\omega) - C(\boldsymbol{\theta}^T + \mathbf{A}\boldsymbol{\eta}^T)\right] \\
&\qquad - (\mathbf{A}^\top \boldsymbol{\phi}(\omega) - \mathbf{b}) \cdot \boldsymbol{\eta}^T - \boldsymbol{\theta}^0 \cdot \boldsymbol{\phi}(\omega) + C(\boldsymbol{\theta}^0) - \textstyle\sum_{t=1}^T \Delta^t \\
&\leq R\big(\boldsymbol{\phi}(\omega)\big) - \boldsymbol{\theta}^0 \cdot \boldsymbol{\phi}(\omega) + C(\boldsymbol{\theta}^0) - \textstyle\sum_{t=1}^T \Delta^t \\
&= D\big(\boldsymbol{\phi}(\omega) \| \boldsymbol{\theta}^0\big) - \textstyle\sum_{t=1}^T \Delta^t
\end{aligned}$$

where the inequality follows by Eq. (2) and because $\mathbf{A}^\top \boldsymbol{\phi}(\omega) \geq \mathbf{b}$ and $\boldsymbol{\eta}^T \geq \mathbf{0}$. □

The protocol above allows a wide range of arbitrage detection strategies. In particular, note that the set of constraints $\mathcal{K}$ can grow over time. For example, there are classes of constraints which are exponentially large (in the number of base events), but which have efficient algorithms for finding the most violated constraint. Even when efficient exact algorithms are not available, we might be able to design approximation algorithms or heuristics which return violated constraints. This approach is known as *constraint generation*. The constraint generation can run in parallel with the prediction market protocol or as the first step of the update of $\boldsymbol{\eta}$. In the next section we work out a detailed example of specific constraints and constraint generation approaches for the conjunction market. In the remainder of this section, we discuss two approaches for updating $\boldsymbol{\eta}$, assuming a fixed set of constraints.

3.3. Projected gradient-descent update

For this update, we assume that the gradient $\nabla_{\boldsymbol{\eta}} \tilde{C}(\boldsymbol{\theta}, \boldsymbol{\eta})$ is Lipschitz-continuous and L is an upper bound on its Lipschitz constant. We consider the t-th round of the protocol, and write $\boldsymbol{\theta}$ for $\boldsymbol{\theta}^t$, $\boldsymbol{\eta}$ for $\boldsymbol{\eta}^{t-1}$ and $\boldsymbol{\eta}'$ for $\boldsymbol{\eta}^t$. Let $\boldsymbol{\mu} = \boldsymbol{p}(\boldsymbol{\theta} + \mathbf{A}\boldsymbol{\eta})$. Let

$$\mathbf{g} = \nabla_{\boldsymbol{\eta}} \tilde{C}(\boldsymbol{\theta}, \boldsymbol{\eta}) = \mathbf{A}^\top \boldsymbol{\mu} - \mathbf{b} \ , \quad \hat{\mathbf{g}} = \min(\mathbf{g}, L\boldsymbol{\eta}) \ ,$$

where the minimum in the definition of $\hat{\mathbf{g}}$ is entry-wise. Note that g_k and $\hat{g}_k$ are negative whenever the constraint k is violated. The projected gradient-descent update is of the form

$$\boldsymbol{\eta}' = (\boldsymbol{\eta} - \mathbf{g}/L)_+ = \boldsymbol{\eta} - \hat{\mathbf{g}}/L$$

where $(\cdot)_+$ denotes the entry-wise positive part. The next theorem shows that this update decreases extended cost whenever some constraints are violated:

THEOREM 3.2. *Let L, $\boldsymbol{\theta}$, $\boldsymbol{\eta}$, $\mathbf{g}$, $\hat{\mathbf{g}}$ be defined as above. Then*

$$\tilde{C}(\boldsymbol{\theta}, \boldsymbol{\eta}') - \tilde{C}(\boldsymbol{\theta}, \boldsymbol{\eta}) \leq -\boldsymbol{\eta} \cdot (\mathbf{g} - \hat{\mathbf{g}}) - \|\hat{\mathbf{g}}\|^2/2L \leq -\|\hat{\mathbf{g}}\|^2/2L \ .$$

PROOF. The proof is standard and follows by plugging the expression for $\boldsymbol{\eta}'$ into the Taylor-series style upper bound

$$\tilde{C}(\boldsymbol{\theta}, \boldsymbol{\eta}') \leq \tilde{C}(\boldsymbol{\theta}, \boldsymbol{\eta}) + (\boldsymbol{\eta}' - \boldsymbol{\eta}) \cdot \nabla_{\boldsymbol{\eta}} \tilde{C}(\boldsymbol{\theta}, \boldsymbol{\eta}) + \tfrac{L}{2} \|\boldsymbol{\eta}' - \boldsymbol{\eta}\|^2 \ . \qquad \square$$

If the Lipschitz constant is initially not known, it is possible to use a strategy along the lines of [Nesterov 2007]: initialize L by a conservative lower bound, and verify that the

bound in the proof of Thm. 3.2 holds at $\boldsymbol{\eta}'$; whenever it does not hold, increase L by a constant factor. The Lipschitz constant will be found with an additive overhead in the running time proportional to the log of the ratio between the Lipschitz constant and the initial lower bound. The calculated Lipschitz constant is reused in consecutive rounds of the protocol, so this overhead is incurred only once (and asymptotically vanishes as the number of trades T increases).

3.4. Coordinate-descent update for sums of LMSR markets

The running time of the projected-gradient update is proportional to the number of non-zero entries of matrix $\mathbf{A}$. As more and more securities are traded and more constraints generated, this can become prohibitively expensive. An alternative is to update one coordinate η_k at a time with a running time proportional to the number of non-zero coefficients in the constraint. Projected gradient descent and Thm. 3.2 discussed in the previous section can be easily adapted to single-coordinate updates. Here we describe an alternative coordinate-descent approach based on optimizing an upper bound similar to Dudík et al.'s PLUMMET algorithm [2007]. Empirically, we have found that this update decreases the cost faster than the Taylor-series bound of Thm. 3.2.

We only treat the independent-markets costs where each group uses LMSR. We consider updating a single coordinate η_k by an amount δ. Let $\mathbf{e}_k$ be the k-th basis vector, and for a fixed $\boldsymbol{\eta}$, $\boldsymbol{\theta}$, let F denote the change in the cost function after the update $\boldsymbol{\eta}' \leftarrow \boldsymbol{\eta} + \delta\mathbf{e}_k$:

$$F(\delta) = \tilde{C}(\boldsymbol{\theta}, \boldsymbol{\eta} + \delta\mathbf{e}_k) - \tilde{C}(\boldsymbol{\theta}, \boldsymbol{\eta}) \ .$$

As before, let $\boldsymbol{\mu} = \boldsymbol{p}(\boldsymbol{\theta} + \mathbf{A}\boldsymbol{\eta})$. Let A_{ik} denote a single entry of $\mathbf{A}$, and $\mathbf{A}_g$ the block with entries in $\mathbb{R}^{g \times \mathcal{K}}$. Using Eq. (1), we can rewrite F as

$$\begin{aligned} F(\delta) &= \sum_{g \in \mathcal{G}} \Big[C_g\big(\boldsymbol{\theta}_g + \mathbf{A}_g(\boldsymbol{\eta} + \delta\mathbf{e}_k)\big) - C_g(\boldsymbol{\theta}_g + \mathbf{A}_g\boldsymbol{\eta}) \Big] - \delta b_k \\ &= \sum_{g \in \mathcal{G}} B_g \ln \left(\sum_{i \in g} \mu_i e^{\delta A_{ik}/B_g} \right) - \delta b_k \ . \end{aligned}$$

We seek to decrease the cost function as much as possible by minimizing F over $\delta \geq -\eta_k$ (this constraint is equivalent to $\eta'_k \geq 0$). We proceed by bounding F from above using the inequality $\log(1 + x) \leq x$:

$$F(\delta) \leq -\delta b_k + \sum_{g \in \mathcal{G}} \sum_{i \in g} B_g \mu_i (e^{\delta A_{ik}/B_g} - 1) \ . \tag{5}$$

We bound $e^{\delta A_{ik}\eta_k/B_g}$ from above using the following piecewise linear bound, which is valid for $|x| \leq M$:

$$e^{\delta x} - 1 \leq \begin{cases} x(e^{\delta M} - 1) \,/\, M & \text{if } x > 0 \\ (-x)(e^{-\delta M} - 1) \,/\, M & \text{if } x < 0 \\ 0 & \text{if } x = 0 \end{cases}$$

To apply this bound, we split the sum in Eq. (5) into two sub-sums corresponding to $A_{ik} > 0$ and $A_{ik} < 0$. Denoting

$$M = \max_{g \in \mathcal{G},\, i \in g} |A_{ik}/B_g| \ , \quad A^+ = \sum_{i \in \mathcal{I}:\ A_{ik} > 0} \mu_i A_{ik} \ , \quad A^- = \sum_{i \in \mathcal{I}:\ A_{ik} < 0} \mu_i (-A_{ik}) \ ,$$

and plugging the piecewise linear upper bound into Eq. (5) yields

$$F(\delta) \leq -\delta b_k + A^+(e^{\delta M} - 1)/M + A^-(e^{-\delta M} - 1)/M \ . \tag{6}$$

Now, we can explicitly minimize the right-hand side of Eq. (6) under the constraint $\delta \geq -\eta_k$. From the first-order optimality conditions, we find that the optimizing δ^* satisfies the following:

(1) if $\left(-b_k + A^+ e^{-\eta_k M} - A^- e^{\eta_k M}\right) > 0$: $\quad \delta^* = -\eta_k$

(2) otherwise: $\quad \delta^* = \begin{cases} \frac{1}{M} \ln\left(\frac{b_k + \sqrt{b_k^2 + 4A^+A^-}}{2A^+}\right) & \text{if } A^+ > 0 \\ \frac{1}{M} \ln\left(-\frac{A^-}{b_k}\right) & \text{if } A^+ = 0 \end{cases}$

Similar to projected-gradient style single-coordinate update, the update above requires only prices that appear with non-zero coefficient in the constraint k, and after the update only these same prices are affected. Thus, the time complexity is proportional to the number of non-zero coefficients in the corresponding constraint (we treat the maximum group size as a constant).

4. CONJUNCTION MARKET

In this section we construct a tractable *conjunction market* over Boolean variables. We begin by describing securities and their grouping, and continue by providing a specific set of equality and inequality constraints.

The outcome space is an n-dimensional Boolean hypercube $\Omega = \{0,1\}^n$ defining *base events*

$$A_j = \{\omega \in \Omega : \omega_j = 1\} \text{ and } A_j^c = \{\omega \in \Omega : \omega_j = 0\} \text{ for } j = 1, \ldots, n.$$

Base events correspond to literals in propositional logic. Let $\mathcal{L} = \{A_1, A_1^c, \ldots, A_n, A_n^c\}$, be the set of all literals. We construct the index set $\mathcal{I}$ by combining literals and symbols $\cap$ and $\cup$ into certain set-theoretic expressions. The payoff function for each security i is defined as $\phi_i(\omega) = \mathbf{1}[\omega \in i]$. Realizability in this case amounts to consistency among prices when viewed as probabilities.

We consider the following security groups:

(1) *base securities*: for each $j = 1, \ldots, n$, we introduce the group $\{A_j,\ A_j^c\}$
(2) *base-pair securities*: for a pair of distinct positive literals $A, B \in \mathcal{L}$, we introduce the group $\{A \cap B,\ A \cap B^c,\ A^c \cap B,\ A^c \cap B^c\}$
(3) *conjunction and disjunction securities*: let $m \geq 3$ and $L_1, \ldots, L_m \in \mathcal{L}$ such that L_j are literals from distinct groups; let L_1^c be the complementary literals; we introduce the group $\{L_1 \cap \cdots \cap L_m,\ L_1^c \cup \cdots \cup L_m^c\}$

The total number of securities is exponential in n, but we do not explicitly keep track of all of them. We initialize the market-maker with base securities only. The remaining security groups are added during the execution of mechanism as traders request price quotes and realize trades. Once the security is added it remains in the set $\mathcal{I}$. Thus, the representation of the market state (ignoring linear constraints) is increasing at most linearly with the number of trades.

When a new security is added, its price needs to be initialized. Base events are typically initialized using some prior information. For elections, we may use poll results or historic averages. For derived events, we can either use prior information (if available), or form a reasonable initial price based on the prices of the component events. Let $\boldsymbol{\mu}$ be the current price vector, and write $\mu[i]$ as an alternative notation for μ_i. We use the following initialization:

— base-pair security $L_1 \cap L_2$, where $L_1, L_2 \in \mathcal{L}$, is initialized as

$$\mu[L_1 \cap L_2] \leftarrow \mu[L_1]\mu[L_2]$$

— when a conjunction security $L_1 \cap \cdots \cap L_m$ and the complementary disjunction are added, we first ensure that all the base-pair securities $L_j \cap L_k$ for $1 \le j < k \le m$ are in the index set $\mathcal{I}$ (e.g., by recursively initializing), and initialize as

$$\mu[L_1 \cap \cdots \cap L_m] \leftarrow \min \left\{ \min_{1 \le j \le m} \mu[L_j], \ \min_{1 \le j < k \le m} \mu[L_j \cap L_k] \right\} \ .$$

4.1. Local consistency constraints

For all base-pair securities and conjunctions we introduce the following constraints to ensure that they are locally consistent with the individual literals appearing in them:

— for a base-pair group $\{A \cap B,\ A \cap B^c,\ A^c \cap B,\ A^c \cap B^c\}$:

$$\mu[A \cap B] + \mu[A \cap B^c] = \mu[A]$$
$$\mu[A \cap B] + \mu[A^c \cap B] = \mu[B]$$

— for a conjunction $L_1 \cap \cdots \cap L_m$ and each of its literals L_j:

$$\mu[L_1 \cap \cdots \cap L_m] \le \mu[L_j] \ .$$

4.2. Clique constraints

Consider a disjunction $L_1 \cup \cdots \cup L_m$. Let $\{L'_1, \ldots, L'_k\} \subseteq \{L_1, \ldots, L_m\}$. For any probability distribution $\mathbb{P}$, we must have (by the union bound and Bonferroni inequalities):

$$\mathbb{P}[L_1 \cup \cdots \cup L_m] \ge \mathbb{P}[L'_1 \cup \cdots \cup L'_k] \ge \left(\sum_{1 \le j \le k} \mathbb{P}[L'_j] \right) - \left(\sum_{1 \le j_1 < j_2 \le k} \mathbb{P}[L'_{j_1} \cap L'_{j_2}] \right) \ .$$

The above inequality gives rise to a *clique constraint*:

$$\mu[L_1 \cup \cdots \cup L_m] \ge \left(\sum_{1 \le j \le k} \mu[L'_j] \right) - \left(\sum_{1 \le j_1 < j_2 \le k} \mu[L'_{j_1} \cap L'_{j_2}] \right) \ .$$

The "clique" here refers to the fact that each index $1, \ldots, m$ can be viewed as a node and we subtract away the probability $\mu[L'_{j_1} \cap L'_{j_2}]$ of the edge between each pair of nodes. The number of clique constraints grows exponentially with the size of conjunction. Rather than including all constraints, we only attempt to find the most violated one, corresponding to the clique that maximizes the value of the right-hand side. For small conjunctions, it is easy to exhaustively enumerate all possibilities. For larger conjunctions, we note that the objective is a non-monotone submodular function and use a constant-factor approximation algorithm based on local search [Feige et al. 2007].

4.3. Spanning-tree constraints

Following Galambos and Simonelli [1996], the probability of a disjunction $L_1 \cup \cdots \cup L_m$ can also be bounded from above as

$$\mathbb{P}[L_1 \cup \cdots \cup L_m] \le \left(\sum_{1 \le j \le m} \mathbb{P}[L_m] \right) - \left(\sum_{(j_1, j_2) \in T} \mathbb{P}[L_{j_1} \cap L_{j_2}] \right)$$

where T contains $m - 1$ pairs describing edges of a spanning tree on nodes $1, \ldots, m$. Replacing $\mathbb{P}$ by μ, we obtain a *spanning-tree constraint* on realizable prices. Finding the tightest upper bound of this form is equivalent to finding the maximum spanning tree over nodes $1, \ldots, m$ with the edge (j_1, j_2) having the weight $\mu[L_{j_1} \cap L_{j_2}]$.

5. EXPERIMENTS

In this section we report on an empirical evaluation of our constraint-generation based market maker (CGMM) on survey data from the 2008 U.S. presidential election. Our implementation uses coordinate descent to detect and eliminate arbitrage opportunities, bringing market prices closer to coherence, and makes use of the Local, Clique, and Tree constraints described in the previous section.

5.1. Data and Methodology

The data set was provided by Wang et al. [2011] and consists of over 300,000 probability forecasts surveyed from 16,000 individual judges prior to the 2008 election.[3] The judges were given questionnaires asking for their probability estimates for base events such as "Obama/McCain to win FL", as well as probabilities for conjunctions or disjunctions of up to three base events. (The survey assumes that the probability of a third candidate winning a state is zero.) By the survey design, all fifty states appear in some conjunction or disjunction, and typically in several. To compare our market maker's prediction performance against LMSR, which cannot scale to 50 states, we also sub-selected a data set of securities over just 10 states, allowing for exhaustive enumeration of all possible outcomes. We selected the 10 states that appeared most frequently in the data set, which were: CO, MA, MI, NY, OR, PA, SD, TX, WA, WV.

Tab. I provides a breakdown of the number of events of different types in each data set. While sub-selecting 10 states biases the events towards smaller conjunctions and disjunctions, we found that the distributions of the judges' estimates were qualitatively similar in both data sets, with spikes near 0 and 1 and a uniform distribution in between.

Table I. Distribution of security types across the small and complete survey data sets.

States	P	$P \cap Q$	$P \cup Q$	$P \cap Q \cap S$	$P \cup Q \cup S$	Total
10	27,754	2,280	2,327	399	387	33,147
50	134,841	57,789	57,789	57,789	57,789	365,997

For our evaluation, we ignore the judge identities in the data and instead treat each individual probability estimate as a single agent that seeks to trade on the corresponding security. The agents are each given the same budget, and are processed in order (after a random permutation). Each agent is processed only once. An agent purchases shares of its desired security until the price reaches its probability estimate, or its budget is exhausted, whichever occurs first. We do not allow short selling: if an agent's estimate lies below the current price then it instead buys the complementary security, which is equivalent to short-selling. In this experimental setup the key parameter is the budget, which controls the degree to which agents are able to move prices towards their estimates.

Once all agents have been processed, we take the final prices as probability estimates and evaluate their stochastic accuracy according to two standard scoring rules. In the following, let A be the event that occurs within a given group, and let $\mu[A]$ be the price (i.e., probability) that the market maker ascribes to A.

— *Log score.* The log score is defined as $\ln \mu[A]$.
— *Quadratic score.* The quadratic score is defined as $-(1 - \mu[A])^2$.

[3]The survey data can be obtained at `www.princeton.edu/guanchun/Election_Data/`.

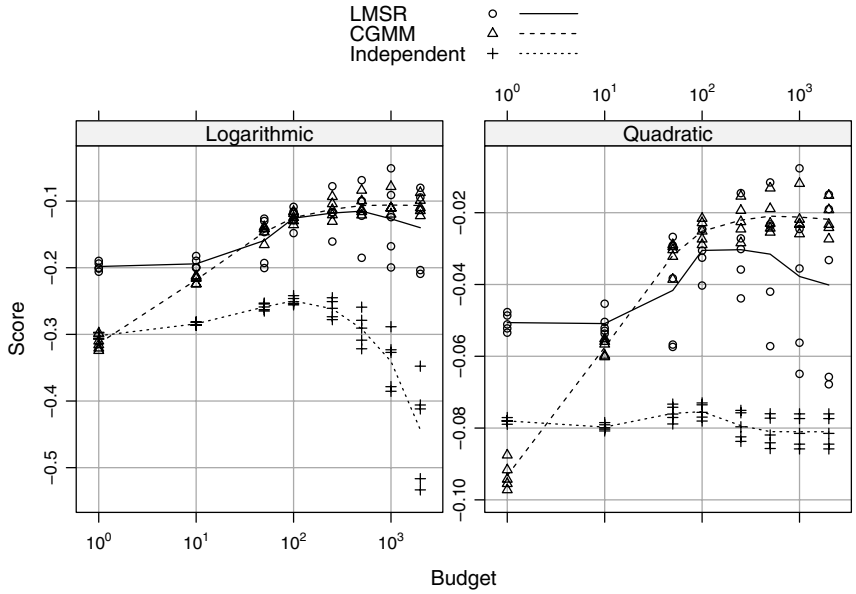

Fig. 1. Forecasting performance of CGMM, using all types of constraints, against the benchmarks of LMSR and independent markets on the small dataset (10 states). For each budget setting, the symbols indicate the individual results for the five permutations. Curves are plotted through the mean score at each budget.

Both of these scoring rules are proper, meaning that they are maximized in expectation by reporting one's true belief. Note that each score is normalized in the sense that it takes on negative values and the maximum possible score is zero. Scoring rules are simply negative loss functions; we use scoring rules to be consistent with the literature on prediction markets [Wang et al. 2011]. The corresponding loss functions are standard in machine learning: the log score corresponds to log loss which is optimized by the maximum-likelihood estimate, while the quadratic score (also known as Brier score) corresponds to squared error which is optimized by the least-squares estimate.

The final score is obtained by taking the average score over all securities that *some agent explicitly purchased*, not all securities that were internally created by the conjunction market. The securities purchased correspond to the events of interest in the election. For each market maker we initialize the prices of base securities to the probabilities of their events according to polls from 5 months prior to the election.

5.2. Evaluation on small data

We first examine forecasting performance on the small data set with 10 states where it is possible to evaluate LMSR. As another benchmark we also consider independent markets for all the securities. Our market maker can be run with various selections of constraint generation schemes. At a minimum, it always uses Local consistency constraints, and beyond these Tree and Clique constraint generators can be used alone or in combination.

We obtained five permutations of the agent data set and evaluated the market makers on all five for a range of budget settings. Fig. 1 displays the Log and Quadratic scores for the two benchmarks as well as our market maker using Local, Tree, and Clique constraints in combination. We find that our market maker soundly outperforms independent markets for all budgets except small ones close to $1. We also see that it achieves the highest score for the Quadratic score, while it is competitive with

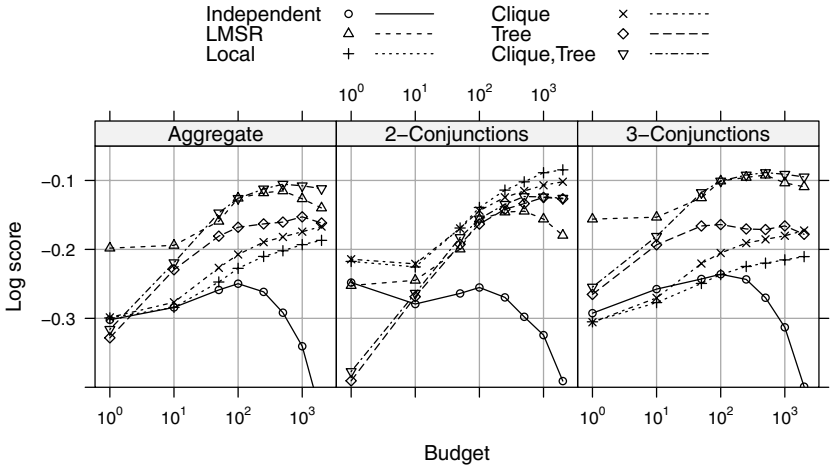

Fig. 2. Forecasting performance according to Log score using different constraint combinations. Each symbol represents an average over five permutations of the data set. The left panel gives the average score over all security types. The middle and right panels give the average score across conjunctions and disjunctions only, of size exactly 2 and 3.

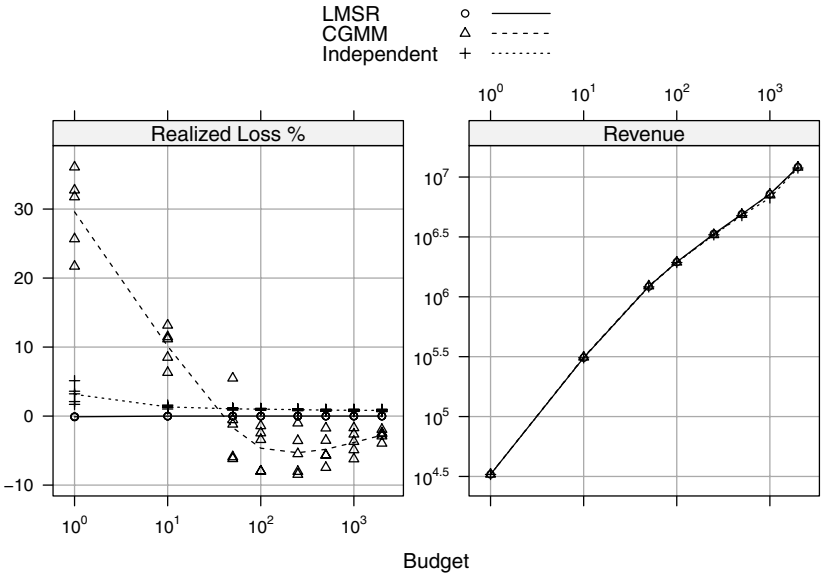

Fig. 3. Realized loss—money paid out to agents minus revenue—as a percent of revenue, together with revenue growth as a function of budget. Each symbol in the left-hand panel corresponds to a permutation of the data set, whereas each symbol in the right-hand panel represents an average over permutations. Curves are drawn through the mean over the permutations at each budget.

LMSR according to Log score. The figure also indicates that CGMM shows little variance in performance compared to LMSR.

Fig. 2 provides insights into the contribution of different types of constraints to forecasting performance. We see an ordering in the contribution of constraints once the budget is large enough. Clique constraints improve performance over Local con-

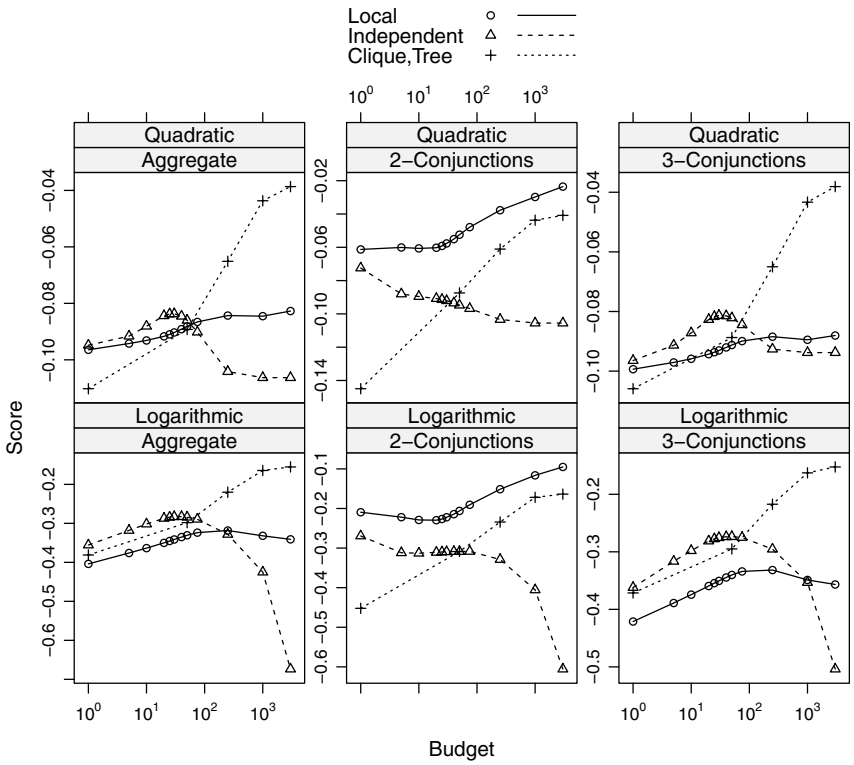

Fig. 4. Forecasting performance of CGMM, using all types of constraints, against independent markets on the complete dataset (50 states). Top panels give Quadratic scores while bottom panels give Log scores. Note that scales for the vertical axes vary between plots.

straints, but not as much as Tree constraints. Notably, it is the combination of the two that gives a substantial boost and makes CGMM competitive with LMSR; one provides upper bounds and the other lower bounds, which makes them complementary. The same ordering is found for the performance on conjunctions and disjunctions of size 3, but this is not the case for those of size 2, where Tree and Clique deter somewhat from the Local constraints. While CGMM performance improves with increasing budget for the range examined here, LMSR and especially independent markets start to suffer when budget is large enough. An explanation for this lies in the mechanics of CGMM as compared to the benchmarks. With independent markets and LMSR, when an agent with large budget moves the price on an event too far towards 0 or 1, it is up to the following agents to correct it. With CGMM, the constraint generation following the purchase will move the price back in line with other security prices.

This explanation is also borne out by Fig. 3. For large enough budgets, CGMM begins to run a gain, which can be attributed to the automated arbitrage (i.e., constraint generation) following aggressive security purchases by agents moving prices to extremes and incoherence. The loss trends for independent markets and LMSR are milder given that these do not perform arbitrage. The revenue growth as a function of budget is analogous for all market makers.

5.3. Evaluation on complete data

We now turn to forecasting performance on the complete data set with 50 states. Because LMSR requires exhaustive enumeration of outcomes, it cannot scale to this level

(indeed, it could not scale beyond 20 states), so independent markets are the sole benchmark. Fig. 4 compares CGMM with Local constraints alone and with all constraint generators (Tree and Clique) against independent markets. We find the same trends as for the small data set. CGMM with Tree and Clique constraints outperforms the benchmark according to both Quadratic and Log scores. The performance of CGMM is more robust to the budget level than independent markets. We again see that Tree and Clique constraints together improve over Local constraints for all securities in aggregate, and for conjunctions/disjunctions of size 3, but Local constraints alone dominate for those of size 2. We conclude from these trends that CGMM performs well with large budget levels, which allow agents to fully incorporate their information into the market while the constraint generation brings extreme prices back in line with other security prices.

6. CONCLUSIONS

In this work we presented an automated market maker based on convex optimization and constraint generation that lies between independent markets and a full combinatorial market. We proved bounds on its loss and proposed a specific implementation for conjunction securities of relevance to domains such as presidential elections. Our evaluation showed that the market maker is competitive with LMSR but retains the ability to scale to exponential outcome spaces. The reason for its promise in practice lies in its modularity: there is substantial freedom to choose the optimization algorithm to eliminate arbitrage opportunities (e.g., projected-gradient descent or coordinate descent), and many possible schemes for effective constraint generation.

In immediate future work, we plan to introduce *threshold constraints* corresponding to threshold securities which capture events of the form "Obama wins at least two Southwest states". Indeed, the most natural election security, "Obama/Romney wins the presidency", corresponds to a weighted threshold constraint. Note that these generalize both conjunctions and disjunctions. To handle such constraints, we see opportunities to draw on so-called cycle constraints [Sontag and Jaakkola 2007] from graphical models as well as more general integer-programming techniques such as Gomory cuts.

Another avenue for future work is to introduce non-myopic agents into the simulations to evaluate the robustness of the different market makers against more intelligent traders. For instance, one could consider agents that update their beliefs based on market prices, agents that hold beliefs over several related securities (such as the judges in our data set), and agents that bet strategically based on their perceived ability to move market prices.

ACKNOWLEDGMENTS

We thank Jake Abernethy, Rafael Frongillo, Dan Osherson, David Rothschild, Rob Schapire, and Arvid Wang.

REFERENCES

ABERNETHY, J., CHEN, Y., AND WORTMAN, J. 2011. An optimization-based framework for automated market-making. In *ACM Conference on Electronic Commerce*.

AGRAWAL, S., WANG, Z., AND YE, Y. 2008. Parimutuel betting on permutations. In *International Workshop on Internet and Network Economics*. 126–137.

CHEN, Y., FORTNOW, L., LAMBERT, N., PENNOCK, D. M., AND WORTMAN, J. 2008a. Complexity of combinatorial market makers. In *ACM Conference on Electronic Commerce*. ACM, New York, NY, USA, 190–199.

CHEN, Y., FORTNOW, L., NIKOLOVA, E., AND PENNOCK, D. M. 2007. Betting on permutations. In *ACM Conference on Electronic Commerce*. 326–335.

CHEN, Y., GOEL, S., AND PENNOCK, D. M. 2008b. Pricing combinatorial markets for tournaments. In *ACM Symposium on Theory of Computing*. ACM, New York, NY, USA, 305–314.

CHEN, Y. AND PENNOCK, D. M. 2007. A utility framework for bounded-loss market makers. In *Conference on Uncertainty in Artificial Intelligence*. 49–56.

DUDÍK, M., PHILLIPS, S. J., AND SCHAPIRE, R. E. 2007. Maximum entropy density estimation with generalized regularization and an application to species distribution modeling. *Journal of Machine Learning Research 8*, 1217–1260.

FEIGE, U., MIRROKNI, V. S., AND VONDRAK, J. 2007. Maximizing non-monotone submodular functions. In *IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society, Washington, DC, USA, 461–471.

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.

GALAMBOS, J. AND SIMONELLI, I. 1996. *Bonferroni-Type Inequalities with Applications*. Springer, New York.

GUO, M. AND PENNOCK, D. M. 2009. Combinatorial prediction markets for event hierarchies. In *International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 201–208.

HANSON, R. D. 2003. Combinatorial information market design. *Information Systems Frontiers 5,* 1, 107–119.

HANSON, R. D. 2007. Logarithmic market scoring rules for modular combinatorial information aggregation. *Journal of Prediction Markets 1,* 1, 1–15.

HANSON, R. D., OPREA, R., AND PORTER, D. 2007. Information aggregation and manipulation in an experimental market. *Journal of Economic Behavior and Organization 60,* 4, 449–459.

NESTEROV, Y. 2007. Gradient methods for minimizing composite objective function. Tech. Rep. 76, Center for Operations Research and Econometrics (CORE), Catholic University of Louvain (UCL).

OTHMAN, A. AND SANDHOLM, T. 2011. Automated market makers that enable new settings: Extending constant-utility cost functions. In *Conference on Auctions, Market Mechanisms and Their Applications*.

PENNOCK, D. M. AND XIA, L. 2011. Price updating in combinatorial prediction markets with Bayesian networks. In *Conference on Uncertainty in Artificial Intelligence*. 581–588.

ROCKAFELLAR, R. T. 1970. *Convex Analysis*. Princeton University Press, Princeton, NJ.

SONTAG, D. AND JAAKKOLA, T. 2007. New outer bounds on the marginal polytope. In *Neural Information Processing Systems*.

WAINWRIGHT, M. J. AND JORDAN, M. I. 2008. Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn. 1*, 1–305.

WANG, G., KULKARNI, S. R., POOR, H. V., AND OSHERSON, D. N. 2011. Aggregating large sets of probabilistic forecasts by weighted coherent adjustment. *Decision Analysis 8*, 128–144.

XIA, L. AND PENNOCK, D. M. 2011. An efficient Monte-Carlo algorithm for pricing combinatorial prediction markets for tournaments. In *International Joint Conference on Artificial Intelligence*. 452–457.