

Efficient Market Making via Convex Optimization, and a Connection to Online Learning *

Jacob Abernethy
EECS Department
University of California, Berkeley
jake@cs.berkeley.edu

Yiling Chen
School of Engineering and Applied Sciences
Harvard University
yiling@eecs.harvard.edu

Jennifer Wortman Vaughan
Computer Science Department
University of California, Los Angeles
jenn@cs.ucla.edu

Working Draft, January 19, 2012¹

Abstract

We propose a general framework for the design of securities markets over combinatorial or infinite state or outcome spaces. The framework enables the design of computationally efficient markets tailored to an arbitrary, yet relatively small, space of securities with bounded payoff. We prove that any market satisfying a set of intuitive conditions must price securities via a convex cost function, which is constructed via conjugate duality. Rather than deal with an exponentially large or infinite outcome space directly, our framework only requires optimization over a convex hull. By reducing the problem of automated market making to convex optimization, where many efficient algorithms exist, we arrive at a range of new polynomial-time pricing mechanisms for various problems. We demonstrate the advantages of this framework with the design of some particular markets. We also show that by relaxing the convex hull we can gain computational tractability without compromising the market institution's bounded budget. Although our framework was designed with the goal of deriving efficient automated market makers for markets with very large outcome spaces, this framework also provides new insights into the relationship between market design and machine learning, and into the complete market setting. Using our framework, we illustrate the mathematical parallels between cost function based markets and online learning and establish a correspondence between cost function based markets and market scoring rules for complete markets.

1 Introduction

Securities markets play a fundamental role in economics and finance. A securities market offers a set of *contingent securities* whose payoffs depend on the future state of the world. For example, an Arrow-Debreu

*Some of the results and text in this paper initially appeared in Chen and Vaughan [15] and Abernethy et al. [2].

¹To appear in ACM Transactions on Economics and Computation. Send comments on this draft to jenn@cs.ucla.edu.

security pays \$1 if a particular state of the world is reached and \$0 otherwise [5, 6]. Consider an Arrow-Debreu security that will pay off in the event that a category 4 or higher hurricane passes through Florida in 2012. A Florida resident who worries about his home being damaged might buy this security as a form of insurance to hedge his risk; if there is a hurricane powerful enough to damage his home, he will be compensated. Additionally, a risk-neutral trader who has reason to believe that the probability of a category 4 or higher hurricane landing in Florida in 2012 is p should be willing to buy this security at any price below p or (short) sell it at any price above p to capitalize his information. For this reason, the market price of the security can be viewed as the traders’ collective estimate of how likely it is that a powerful hurricane will occur. Securities markets thus have dual functions: *risk allocation* and *information aggregation*.

Insurance contracts, options, futures, and many other financial derivatives are examples of contingent securities. A securities market primarily focused on information aggregation is often referred to as a *prediction market*. The forecasts of prediction markets have proved to be accurate in a variety of domains [8, 42, 60]. While our work builds on ideas from prediction market design [4, 15, 49], our framework can be applied to any contingent securities.

A securities market is said to be *complete* if it offers at least $|\mathcal{O}| - 1$ linearly independent securities over a set \mathcal{O} of mutually exclusive and exhaustive states of the world, which we refer to as *outcomes* [5, 6, 45]. For example, a prediction market with n Arrow-Debreu securities for n outcomes is complete. In a complete securities market without transaction fees, a trader may bet on any combination of the securities, allowing him to hedge any possible risk he may have. It is generally assumed that the trader may *short sell* a security, betting against the given outcome; in a market with short selling, the n th security is not strictly necessary, as a trader can substitute the purchase of this security by short selling all others. Furthermore, traders can change the market prices to reflect any valid probability distribution over the outcome space, allowing them to reveal any belief. Completeness therefore provides expressiveness for both risk allocation and information aggregation.

Unfortunately, completeness is not always achievable. In many real-world settings, the outcome space is exponentially large or even infinite. For instance, a competitive race between n athletes results in an outcome space of $n!$ rank orders, while the future price of a stock has an infinite outcome space, namely $\mathbb{R}_{\geq 0}$. In such situations operating a complete securities market is not practical for two reasons: (a) humans are notoriously bad at estimating small probabilities and (b) it is computationally intractable to manage such a large set of securities. Instead, it is natural to offer a smaller set of structured securities. For example, rather than offer a security corresponding to each rank ordering, in *pair betting* a market institution offers securities of the form “\$1 if candidate A beats candidate B” [17, 18]. There has been a surge of recent research examining the tractability of running standard prediction market mechanisms (such as the popular Logarithmic Market Scoring Rule (LMSR) market maker [32]) over combinatorial outcome spaces by limiting the space of available securities [51]. While this line of research has led to a few positive results [3, 16, 19, 31], it has led more often to hardness results [16, 18] or to markets with undesirable properties such as unbounded loss of the market institution [24].

In this paper, we propose a general framework to design *automated market makers* for securities markets. An automated market maker is a market institution that adaptively sets prices for each security and is always willing to accept trades at these prices. Unlike previous research aimed at finding a space of securities that can be efficiently priced using an existing market maker like LMSR, we start with an arbitrary space of securities and design a *new* market maker tailored to this space. Our framework is therefore very general and includes existing market makers for complete markets, such as the LMSR and Quad-SCPM [4], as special cases.

We take an axiomatic approach. Given a relatively small space of securities with bounded payoff, we

define a set of intuitive conditions that a reasonable market maker should satisfy. We prove that a market maker satisfying these conditions must price securities via a convex potential function (the *cost function*), and that the space of reachable security prices must be precisely the convex hull of the payoff vectors for each outcome (that is, the set of vectors, one per outcome, denoting the payoff for each security if that outcome occurs). We then incorporate ideas from online convex optimization [34, 54] to define a convex cost function in terms of an optimization over this convex hull; the vector of prices is chosen as the optimizer of this convex objective. With this framework, instead of dealing with the exponentially large or infinite outcome space, we only need to deal with the lower-dimensional convex hull. The problem of automated market making is reduced to the problem of convex optimization, for which we have many efficient techniques to leverage.

To demonstrate the advantages of our framework, we provide two new computationally efficient markets. The first market can efficiently price *subset bets* on permutations, which are known to be #P-hard to price using LMSR [18]. The second market can be used to price bets on the landing location of an object on a sphere. For situations where the convex hull cannot be efficiently represented, we show that we can relax the convex hull to gain computational tractability without compromising the market maker’s bounded budget. This allows us to provide a computationally efficient market maker for the aforementioned pair betting, which is also known to be #P-hard to price using LMSR [18].

Although our framework was designed with the goal of deriving novel, efficient automated market makers for markets with very large outcome spaces, this framework also provides new insights into the relationship between market design and machine learning, and into the complete market setting. With our framework, we illustrate the mathematical parallels between cost function based markets and online learning, and establish a correspondence between cost function based markets and market scoring rules for complete markets.

Roadmap of the paper The rest of the paper is organized as follows. We begin in Section 2 with a review of the relevant literature on automated market makers and prediction market design. In Section 3 we describe the problem of market design for large outcome spaces, discuss the difficulties inherent to this problem, and introduce our axiomatic approach. In Section 4 we give a detailed framework for constructing pricing mechanisms based on convex optimization and conjugate duality. We give a couple of examples of efficient duality-based cost function market makers in Section 5. In Section 6 we consider the computational issues associated with our framework, and show how the proposed convex optimization problem can be relaxed to gain tractability without increasing the worst-case loss of the market maker. We illustrate the mathematical parallels between our framework and online learning in Section 7. Finally, in Section 8, we describe how our framework can be used to establish a correspondence between cost function based markets and market scoring rules for complete markets.

2 Background and Related Work

Automated market makers for complete markets are well studied in both economics and finance. Our work builds on the literature on cost function based markets [14, 32, 33]. A simple cost function based market maker offers $|\mathcal{O}|$ Arrow-Debreu securities, each corresponding to a potential outcome. The market maker determines how much each security should cost using a differentiable *cost function*, $C : \mathbb{R}^{|\mathcal{O}|} \rightarrow \mathbb{R}$, which is simply a potential function specifying the amount of money currently wagered in the market as a function of the number of shares of each security that have been purchased. If q_o is the number of shares of security o currently held by traders, and a trader would like to purchase a bundle of r_o shares for each security $o \in \mathcal{O}$ (where each r_o could be positive, representing a purchase, zero, or even negative, representing a sale), the

trader must pay $C(\mathbf{q} + \mathbf{r}) - C(\mathbf{q})$ to the market maker. The instantaneous price of security σ (that is, the price per share of an infinitesimal portion of a security) is then $\partial C(\mathbf{q})/\partial q_\sigma$, and is denoted $p_\sigma(\mathbf{q})$.

One example of a cost function based market that has received considerable attention is Hanson’s Logarithmic Market Scoring Rule (LMSR) [14, 32, 33]. The cost function of the LMSR is

$$C(\mathbf{q}) = b \log \sum_{\sigma \in \mathcal{O}} e^{q_\sigma/b}, \quad (1)$$

where $b > 0$ is a parameter of the market controlling the rate at which prices change. The corresponding price function for each security σ is

$$p_\sigma(\mathbf{q}) = \frac{\partial C(\mathbf{q})}{\partial q_\sigma} = \frac{e^{q_\sigma/b}}{\sum_{\sigma' \in \mathcal{O}} e^{q_{\sigma'}/b}}. \quad (2)$$

It is well known that the monetary loss of an automated market maker using LMSR is upperbounded by $b \log |\mathcal{O}|$. Additionally, the LMSR satisfies several other desirable properties, which are discussed in more detail in Section 3.1.

When $|\mathcal{O}|$ is large or infinite, calculating the cost of a purchase becomes intractable in general. Recent research on automated market makers for large outcome spaces has focused on restricting the allowable securities over a combinatorial outcome space and examining whether LMSR prices can be computed efficiently in the restricted space. If the outcome space contains $n!$ rank orders of n competing candidates, it is #P-hard to price *pair bets* (securities of the form “\$1 if and only if candidate A beats candidate B”) or *subset bets* (for example, “\$1 if one of the candidates in subset C finishes at position k ”) using LMSR on the full set of permutations [18]. If the outcome space contains 2^n Boolean values of n binary base events, it is #P-hard to price securities on conjunctions of any two base events (for example, “\$1 if and only if a Democrat wins Florida and Ohio”) using LMSR [18]. This line of research has led to some positive results when the uncertain event enforces particular structure on the outcome space. In particular, for a single-elimination tournament of n teams, securities such as “\$1 if and only if team A wins a k th round game” and “\$1 if and only if team A beats team B given they face off” can be priced efficiently using LMSR [19]. The tractability of these securities is due to a structure-preserving property — the market probability can be represented by a Bayesian network and price updating does not change the structure of the network. Pennock and Xia [52] significantly generalized this result and characterize all structure-preserving securities. For a taxonomy tree on some statistic where the value of the statistic of a parent node is the sum of those of its children, securities such as “\$1 if and only if the value of the statistic at node A belongs to $[x, y]$ ” can be priced efficiently using LMSR [31].

One approach to combat the computational intractability of pricing over combinatorial spaces is to approximate the market prices using sampling techniques. Yahoo!’s Predictalot², a play-money combinatorial prediction market for the NCAA Men’s Basketball playoff, allows traders to bet on almost any combination of the 2^{63} outcomes of the tournament. Predictalot is based on LMSR. Instead of calculating the exact prices for securities, it uses importance sampling to approximate the prices. Xia and Pennock [61] devised a Monte-Carlo algorithm that can efficiently compute the price of any security in disjunctive or conjunctive normal form with guaranteed error bounds. However, using sampling techniques brings a new problem to pricing. The sampling algorithm in general won’t give the same prices if quoted twice, even if the market status remains the same. Because of this, traders can exploit the market to make a profit, which increases the loss of the market maker.

²<http://labs.yahoo.com/project/336>

In this paper, we take a drastically different approach to combinatorial market design. Instead of searching for supportable spaces of securities for existing market makers, we design new market makers tailored to any security space of interest and with desirable theoretical properties. Additionally, rather than requiring that securities have a fixed (e.g., \$1) payoff when the underlying event happens, we allow more general contingent securities with arbitrary, efficiently computable and bounded payoffs.

Our approach makes use of powerful techniques from convex optimization. Agrawal et al. [4] and Peters et al. [53] also use convex optimization for automated market making. One major difference is that they only consider complete markets, while we consider markets with an arbitrary set of securities. They consider the setting in which traders submit limit orders, and formulate a convex optimization problem that can be solved by the market institution in order to decide what quantity of orders to accept. While formulating the problem in terms of limit orders leads to a syntactically different problem, their mechanisms can be turned into equivalent cost function based market makers. Agrawal et al. [4] show that their mechanisms can be formulated as a risk minimization problem with an associated penalty function. Mathematically the penalty function plays a similar role as the conjugate function R in our framework, but they do not explicitly make a connection with conjugate duality.

This paper focuses on cost function based market makers. It is worth noting that there are other market mechanisms, with different properties, designed for securities markets. For complete markets, Dynamic Parimutuel Markets [44, 50] also use a cost function to price securities, however the securities are parimutuel bets whose future payoff is not fixed a priori, but depends on the market activities. Brahma et al. [10] and Das and Magdon-Ismail [21] design Bayesian learning market makers that maintain a belief distribution and update it based on the traders’ behavior. Call markets have been studied to trade securities over combinatorial spaces. In a call market, participants submit limit orders and the market institution determines what orders to accept or reject. Researchers have studied the computational complexity of operating call markets for both permutation [3, 16, 26] and Boolean [22] combinatorics.

Related work on online learning and related work on market scoring rules are discussed in Sections 7 and 8 respectively.

3 An Axiomatic Approach to Market Design

In this work, we are primarily interested in a market-design scenario in which the outcome space \mathcal{O} is exponentially large, or even infinite, making it infeasible to run a complete market; not only is it generally intractable for the market maker to price an exponential number of securities, but it is notoriously difficult for human traders to reason about the probabilities of so many individually unlikely outcomes. To address both of these problems, we restrict the market maker to offer a menu of only K securities for some reasonably-sized K . These securities will be designed by the market maker and one can interpret each security as corresponding to some “interesting” or “useful” query that we might like to make about the future outcome. For example, if a set of players compete in a tournament, the market maker can offer a security for every question of the form “does player X survive beyond round Y ?”

We assume that the payoff of each security, clearly depending on the future outcome σ , can be described by an arbitrary but efficiently-computable function $\rho : \mathcal{O} \rightarrow \mathbb{R}_{\geq 0}^K$; if a trader purchases a share of security i and the true outcome is σ , then the trader is paid $\rho_i(\sigma)$. We call such a security space *complex*. The complete security space is a special case of a complex security space in which $K = |\mathcal{O}|$ and for each $i \in \{1, \dots, K\}$, $\rho_i(\sigma)$ equals 1 if σ is the i th outcome and 0 otherwise. The markets we design enable traders to purchase arbitrary *security bundles* $\mathbf{r} \in \mathbb{R}^K$. A negative element of \mathbf{r} encodes a sale of such a security. The payoff for \mathbf{r} upon outcome σ is exactly $\rho(\sigma) \cdot \mathbf{r}$, where $\rho(\sigma)$ denotes the vector of payoffs for each security for outcome

\mathfrak{o} . Let us define $\rho(\mathcal{O}) := \{\rho(\mathfrak{o}) \mid \mathfrak{o} \in \mathcal{O}\}$.

The first step in the design of automated market makers for complex security spaces is to determine an appropriate set of properties that we would like such market makers to satisfy. To build intuition about which properties might be desirable, we first step back and consider what it is that makes a market maker like LMSR a good choice for complete markets.

3.1 What Makes A Market Maker Reasonable?

Consider the cost function associated with the Logarithmic Market Scoring Rule (Equation 1) and the corresponding instantaneous price functions (Equation 2). This cost function and the resulting market satisfy several natural properties that make LMSR a “reasonable” choice.

1. The cost function is differentiable everywhere. As a result, an instantaneous price $p_{\mathfrak{o}}(\mathbf{q}) = \partial C(\mathbf{q}) / \partial q_{\mathfrak{o}}$ can always be obtained for the security associated with any outcome \mathfrak{o} , regardless of the current quantity vector \mathbf{q} .
2. The market incorporates information from the traders, in the sense that the purchase of a security corresponding to outcome \mathfrak{o} causes $p_{\mathfrak{o}}$ to increase.
3. The market does not provide explicit opportunities for arbitrage. Since instantaneous prices are never negative, traders are never paid to obtain securities. Additionally, the sum of the instantaneous prices of the securities is always 1. If the prices summed to something less than (respectively, greater than) 1, a trader could purchase (respectively, short sell) small equal quantities of each security for a guaranteed profit. This is prevented.

In addition to preventing arbitrage, these properties also ensure that prices can be interpreted naturally as probabilities, representing the market’s current estimate of the distribution over outcomes.

4. The market is *expressive* in the sense that a trader with sufficient funds is always able to set the market prices to reflect his beliefs about the probability of each outcome.³

As described in Section 2, previous research on cost function based markets for combinatorial outcome spaces has focused on developing algorithms to efficiently implement or approximate LMSR pricing [18, 19, 31]. Because of this, there has been no need to explicitly extend these properties to complex markets; the properties hold automatically for any implementation of LMSR. This is no longer the case when our goal is to design new markets tailored to custom sets of securities.

To gain intuition about what makes an arbitrary complex market “reasonable,” let us begin by considering the example of *pair betting* [17, 18]. Suppose our outcome space consists of rankings of a set of n competitors, such as n horses in a race. The outcome of such a race is a permutation $\pi : [n] \rightarrow [n]$, where $[n]$ denotes the set $\{1, \dots, n\}$, and $\pi(i)$ is the final position of i , with $\pi(i) = 1$ being best. A typical market for this setting might offer n Arrow-Debreu securities, with the i th security paying off if and only if $\pi(i) = 1$. Additionally, there might be separate, *independent* markets allowing bets on horses to place (come in first or second) or show (come in first, second, or third). However, running independent markets for sets of outcomes with clear correlations is wasteful in that information revealed in one market does not automatically propagate to the others. Instead, suppose that we would like to define a set of securities that allow traders to make arbitrary *pair bets*; that is, for every i, j , a trader can purchase a security which pays out \$1 whenever $\pi(i) < \pi(j)$. What properties would make a market for pair bets reasonable?

³Othman et al. [49] introduced a similar property for complete markets, which they called *surjectivity*.

The first two properties described above have straight-forward interpretations in this setting. We would still like the instantaneous price of each security to be well-defined at all times; intuitively, the instantaneous price of the security for $\pi(i) < \pi(j)$ should represent the traders' collective belief about the probability that horse i finishes ahead of horse j . Call this price $p_{i,j}$. We would still like the market to incorporate information, in the sense that buying the security corresponding to $\pi(i) < \pi(j)$ should never cause the price $p_{i,j}$ to drop.

The remaining two properties are more tricky to quantify. Intuitively, these properties require us to define a set of constraints over the prices achievable in the market (to prevent arbitrage), and to ensure that any prices reflecting consistent beliefs about the distribution over outcomes can be achieved (for expressiveness). One can come up with various logical constraints that prices should satisfy. For example, $p_{i,j}$ must be nonnegative at all times for all i and j , and $p_{i,j} + p_{j,i}$ must always equal 1 since exactly one of the two securities corresponding to $\pi(i) < \pi(j)$ and $\pi(j) < \pi(i)$ respectively will pay out \$1. Similar reasoning gives us the additional constraint that for all i, j , and k , $p_{i,j} + p_{j,k} + p_{k,i}$ must be at least 1 and no more than 2. But are these constraints enough to prevent arbitrage? Are they too strong to allow the expression of arbitrary consistent beliefs?

In general, this type of ad hoc reasoning can lead us to many apparently reasonable constraints, but does not yield an algorithm to determine whether or not we have generated the full set of constraints necessary to prevent arbitrage, and cannot be applied easily to more complicated security spaces. We address this problem in the next section. We start by formalizing the desirable market properties described above in the context of complex markets. We then provide a precise mathematical characterization of all cost functions that satisfy these properties.

3.2 An Axiomatic Characterization of Complex Markets

We are now ready to formalize a set of conditions or axioms that one might expect a market to satisfy, and show that these conditions lead to some natural mathematical restrictions on the costs of security bundles. (We consider relaxations of these conditions in Section 6.) We do not presuppose a cost function based market. However, we show that the use of a convex cost function is *necessary* given the assumption of path independence on the security purchases.

3.2.1 Path Independence and the Use of Cost Functions

Imagine a sequence of traders entering the marketplace and purchasing security bundles. Let $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \dots$ be the sequence of security bundles purchased. After $t - 1$ such purchases, the t -th trader should be able to enter the marketplace and query the market maker for the cost of arbitrary bundles. The market maker must be able to furnish a cost, denoted $\text{Cost}(\mathbf{r}|\mathbf{r}_1, \dots, \mathbf{r}_{t-1})$, for any bundle \mathbf{r} given a previous trade sequence $\mathbf{r}_1, \dots, \mathbf{r}_{t-1}$. If the trader chooses to purchase \mathbf{r}_t at a cost of $\text{Cost}(\mathbf{r}_t|\mathbf{r}_1, \dots, \mathbf{r}_{t-1})$, the market maker may update the costs of each bundle accordingly. Our first condition requires that the cost of acquiring a bundle \mathbf{r} must be the same regardless of how the trader splits up the purchase.

Condition 1 (Path Independence). *For any \mathbf{r}, \mathbf{r}' , and \mathbf{r}'' such that $\mathbf{r} = \mathbf{r}' + \mathbf{r}''$, for any $\mathbf{r}_1, \dots, \mathbf{r}_t$,*

$$\text{Cost}(\mathbf{r}|\mathbf{r}_1, \dots, \mathbf{r}_t) = \text{Cost}(\mathbf{r}'|\mathbf{r}_1, \dots, \mathbf{r}_t) + \text{Cost}(\mathbf{r}''|\mathbf{r}_1, \dots, \mathbf{r}_t, \mathbf{r}').$$

Path independence helps to reduce both arbitrage opportunities and the strategic play of traders, as traders need not reason about the optimal path leading to some target position. However, it is worth pointing out that there are interesting markets that do not satisfy this condition, such as the continuous double auction

and the market maker for continuous double auctions considered by Brahma et al. [10] and Das and Magdon-Ismail [21]. These markets do not fall into our framework and deserve separate treatment.

It turns out that the path independence alone implies that prices can be represented by a cost function C , as illustrated in the following theorem.

Theorem 1. *Under Condition 1, there exists a cost function $C : \mathbb{R}^K \rightarrow \mathbb{R}$ such that we may always write*

$$\text{Cost}(\mathbf{r}_t | \mathbf{r}_1, \dots, \mathbf{r}_{t-1}) = C(\mathbf{r}_1 + \dots + \mathbf{r}_{t-1} + \mathbf{r}_t) - C(\mathbf{r}_1 + \dots + \mathbf{r}_{t-1}).$$

Proof. Let $C(\mathbf{q}) := \text{Cost}(\mathbf{q} | \emptyset)$. Clearly $C(\mathbf{0}) = \text{Cost}(\mathbf{0} | \emptyset) = 0$. We will show, via induction on t , that for any t and any bundle sequence $\mathbf{r}_1, \dots, \mathbf{r}_t$,

$$\text{Cost}(\mathbf{r}_t | \mathbf{r}_1, \dots, \mathbf{r}_{t-1}) = C(\mathbf{r}_1 + \dots + \mathbf{r}_{t-1} + \mathbf{r}_t) - C(\mathbf{r}_1 + \dots + \mathbf{r}_{t-1}). \quad (3)$$

When $t = 1$, this holds trivially. Assume that Equation 3 holds for all bundle sequences of any length $t \leq T$. By Condition 1,

$$\begin{aligned} & \text{Cost}(\mathbf{r}_{T+1} | \mathbf{r}_1, \dots, \mathbf{r}_T) \\ &= \text{Cost}(\mathbf{r}_{T+1} + \mathbf{r}_T | \mathbf{r}_1, \dots, \mathbf{r}_{T-1}) - \text{Cost}(\mathbf{r}_T | \mathbf{r}_1, \dots, \mathbf{r}_{T-1}) \\ &= C\left(\mathbf{r}_{T+1} + \mathbf{r}_T + \sum_{t=1}^{T-1} \mathbf{r}_t\right) - C\left(\sum_{t=1}^{T-1} \mathbf{r}_t\right) - \left(C\left(\mathbf{r}_T + \sum_{t=1}^{T-1} \mathbf{r}_t\right) - C\left(\sum_{t=1}^{T-1} \mathbf{r}_t\right)\right) \\ &= C\left(\sum_{t=1}^{T+1} \mathbf{r}_t\right) - C\left(\sum_{t=1}^T \mathbf{r}_t\right), \end{aligned}$$

and we see that Equation 3 holds for $t = T + 1$ too. \square

With this theorem in mind, we drop the cumbersome $\text{Cost}(\mathbf{r} | \mathbf{r}_1, \dots, \mathbf{r}_t)$ notation from now on, and write the cost of a bundle \mathbf{r} as $C(\mathbf{q} + \mathbf{r}) - C(\mathbf{q})$, where $\mathbf{q} = \mathbf{r}_1 + \dots + \mathbf{r}_t$ is the vector of previous purchases.

3.2.2 Formalizing the Properties of a Reasonable Market

Recall that one of the functions of a securities market is to aggregate traders' beliefs into an accurate prediction. Each trader may have his own (potentially secret) information about the future, which we represent as a distribution $\mathbf{p} \in \Delta_{|\mathcal{O}|}$ over the outcome space, where $\Delta_n = \{\mathbf{x} \in \mathbb{R}_{\geq 0}^n : \sum_{i=1}^n x_i = 1\}$, the n -simplex. The pricing mechanism should therefore incentivize the traders to reveal \mathbf{p} , but simultaneously avoid providing arbitrage opportunities. Towards this goal, we now revisit the relevant properties of LMSR discussed in Section 3.2, and show how the ideas behind each of these properties can be extended to the complex market setting, yielding four additional conditions on our pricing mechanism.

The first condition ensures that the gradient of C is always well-defined. If we imagine that a trader can buy or sell an arbitrarily small bundle, we would like the cost of buying and selling an infinitesimal quantity of any particular bundle to be the same. If $\nabla C(\mathbf{q})$ is well-defined, it can be interpreted as a vector of instantaneous prices for each security, with $\partial C(\mathbf{q}) / \partial q_i$ representing the price per share of an infinitesimal amount of security i . Additionally, we can interpret $\nabla C(\mathbf{q})$ as the traders' current estimates of the expected payoff of each security, in the same way that $\partial C(\mathbf{q}) / \partial q_o$ was interpreted as the probability of outcome o when considering the complete security space.

Condition 2 (Existence of Instantaneous Prices). C is continuous and differentiable everywhere on \mathbb{R}^K .

The next condition encompasses the idea that the market should react to trades in a sensible way in order to incorporate the private information of the traders. In particular, it says that the purchase of a security bundle \mathbf{r} should never cause the market to lower the price of \mathbf{r} . This condition is closely related to incentive compatibility for a myopic trader. It is equivalent to requiring that a trader with a distribution $\mathbf{p} \in \Delta_{|\mathcal{O}|}$ can never find it simultaneously profitable (in expectation) to buy a bundle \mathbf{r} or to buy the bundle $-\mathbf{r}$. In other words, there can not be more than one way to express one's information.

Condition 3 (Information Incorporation). For any \mathbf{q} and $\mathbf{r} \in \mathbb{R}^K$, $C(\mathbf{q}+2\mathbf{r}) - C(\mathbf{q}+\mathbf{r}) \geq C(\mathbf{q}+\mathbf{r}) - C(\mathbf{q})$.

The no arbitrage condition states that it is never possible for a trader to purchase a security bundle \mathbf{r} and receive a positive profit regardless of the outcome. Without this property, the market maker would occasionally offer traders a chance to obtain a guaranteed profit, which is clearly suboptimal in terms of the market maker's loss. However, we do consider the relaxation of this property in Section 6.

Condition 4 (No Arbitrage). For all $\mathbf{q}, \mathbf{r} \in \mathbb{R}^K$, there exists an $\mathfrak{o} \in \mathcal{O}$ such that $C(\mathbf{q}+\mathbf{r}) - C(\mathbf{q}) \geq \mathbf{r} \cdot \rho(\mathfrak{o})$.

Finally, the expressiveness condition specifies that any trader can set the market prices to reflect his beliefs, within any ϵ error, about the expected payoffs of each security if arbitrarily small portions of shares may be purchased. The ϵ approximation factor is necessary because the trader's beliefs may only be expressible in the limit.

Condition 5 (Expressiveness). For any $\mathbf{p} \in \Delta_{|\mathcal{O}|}$ we write $\mathbf{x}^{\mathbf{p}} := \mathbb{E}_{\mathfrak{o} \sim \mathbf{p}}[\rho(\mathfrak{o})]$. Then for any $\mathbf{p} \in \Delta_{|\mathcal{O}|}$ and any $\epsilon > 0$ there is some $\mathbf{q} \in \mathbb{R}^K$ for which $\|\nabla C(\mathbf{q}) - \mathbf{x}^{\mathbf{p}}\| < \epsilon$.

Having formalized our set of conditions, we must now address the question of how to determine whether or not these conditions are satisfied for a particular cost function C . The following theorem precisely characterizes the set of all cost functions that satisfy these conditions. The statement and proof require the use of a few pieces of terminology; for more on why this is necessary, see the note in Section 4. In particular, the *relative boundary* of a convex set S is its boundary in the ‘‘ambient’’ dimension of S . For example, if we consider the n -dimensional probability simplex $\Delta_n := \{\mathbf{x} \in \mathbb{R}^n : \sum_i x_i = 1, x_i \geq 0 \forall i\}$, then the relative boundary of Δ_n is the set $\{\mathbf{x} \in \Delta_n : x_i = 0 \text{ for some } i\}$. We use $\text{relint}(S)$ to refer to the *relative interior* of a convex set S , which is the set S minus all of the points on the relative boundary. The interior of a square in 3-dimensional space is empty, but the relative interior is not. We will use $\text{closure}(S)$ to refer to the closure of S , the smallest closed set containing all of the limit points of S . For any subset S of \mathbb{R}^d , let $\mathcal{H}(S)$ denote the convex hull of S . Recall that $\rho(\mathcal{O}) := \{\rho(\mathfrak{o}) | \mathfrak{o} \in \mathcal{O}\}$. Note that $\text{closure}(\mathcal{H}(\rho(\mathcal{O})))$ may not be equal to $\mathcal{H}(\rho(\mathcal{O}))$ if we consider infinite spaces of outcomes or contracts.

Theorem 2. Under Conditions 2-5, C must be convex with

$$\text{closure}(\{\nabla C(\mathbf{q}) : \mathbf{q} \in \mathbb{R}^K\}) = \text{closure}(\mathcal{H}(\rho(\mathcal{O}))).$$

Proof. We first prove convexity. Assume C is non-convex somewhere. Then there must exist some \mathbf{q} and \mathbf{r} such that $C(\mathbf{q}) > (1/2)C(\mathbf{q}+\mathbf{r}) + (1/2)C(\mathbf{q}-\mathbf{r})$. This means $C(\mathbf{q}+\mathbf{r}) - C(\mathbf{q}) < C(\mathbf{q}) - C(\mathbf{q}-\mathbf{r})$, which contradicts Condition 3, so C must be convex.

To prove the equality, we will establish the following two containments:

$$\text{relint}(\mathcal{H}(\rho(\mathcal{O}))) \subseteq \{\nabla C(\mathbf{q}) : \mathbf{q} \in \mathbb{R}^K\} \subseteq \text{closure}(\mathcal{H}(\rho(\mathcal{O}))).$$

We now prove the second \subseteq statement. Notice that Condition 2 trivially guarantees that $\nabla C(\mathbf{q})$ is well-defined for any \mathbf{q} . To see that $\{\nabla C(\mathbf{q}) : \mathbf{q} \in \mathbb{R}^K\} \subseteq \text{closure}(\mathcal{H}(\boldsymbol{\rho}(\mathcal{O})))$, let us assume there exists some \mathbf{q}' for which $\nabla C(\mathbf{q}') \notin \text{closure}(\mathcal{H}(\boldsymbol{\rho}(\mathcal{O})))$. This can be reformulated in the following way: There must exist some halfspace, defined by a normal vector \mathbf{r} , that separates $\nabla C(\mathbf{q}')$ from every member of $\boldsymbol{\rho}(\mathcal{O})$. More precisely

$$\nabla C(\mathbf{q}') \notin \text{closure}(\mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))) \iff \exists \mathbf{r} \forall \mathbf{o} \in \mathcal{O} : \nabla C(\mathbf{q}') \cdot \mathbf{r} < \boldsymbol{\rho}(\mathbf{o}) \cdot \mathbf{r}.$$

On the other hand, letting $\mathbf{q} := \mathbf{q}' - \mathbf{r}$, we see by convexity of C that $C(\mathbf{q} + \mathbf{r}) - C(\mathbf{q}) \leq \nabla C(\mathbf{q}') \cdot \mathbf{r}$. Combining these last two inequalities, we see that the price of bundle \mathbf{r} purchased with history \mathbf{q} is always smaller than the payoff for *any* outcome. This implies that there exists some arbitrage opportunity, contradicting Condition 4.

We finish by proving the first \subseteq statement. Notice that $\mathcal{H}(\boldsymbol{\rho}(\mathcal{O})) = \{\mathbb{E}_{\mathbf{o} \sim \mathbf{p}}[\boldsymbol{\rho}(\mathbf{o})] \mid \mathbf{p} \in \Delta_{|\mathcal{O}|}\}$ which is convex, and recall that the set of derivatives $\{\nabla C(\mathbf{q}) : \mathbf{q} \in \mathbb{R}^K\}$ of any convex function C must form a convex set. The statement of Condition 5 is equivalent to the statement that every element $\mathbf{x}^{\mathbf{p}} \in \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$ is a limit point of the set $\{\nabla C(\mathbf{q}) : \mathbf{q} \in \mathbb{R}^K\}$. But we have just established that $\{\nabla C(\mathbf{q}) : \mathbf{q} \in \mathbb{R}^K\} \subseteq \text{closure}(\mathcal{H}(\boldsymbol{\rho}(\mathcal{O})))$ and thus the only case where $\mathbf{x}^{\mathbf{p}}$ does not equal $\nabla C(\mathbf{q})$ for some \mathbf{q} is when $\mathbf{x}^{\mathbf{p}}$ lies on the relative boundary of $\mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$. \square

What we have arrived at from the set of proposed conditions is that (a) a pricing mechanism can always be described precisely in terms of a convex cost function C and (b) the set of reachable prices of a mechanism, that is the set $\{\nabla C(\mathbf{q}) : \mathbf{q} \in \mathbb{R}^K\}$, must be *identically* the convex hull of the payoff vectors for each outcome $\mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$ except possibly differing at the relative boundary of $\mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$. For complete markets, this would imply that the set of achievable prices should be the convex hull of the n standard basis vectors. Indeed, this comports exactly with the natural assumption that the vector of security prices in complete markets should represent a probability distribution, or equivalently that it should lie in the n -simplex [4].

4 Designing the Cost Function via Conjugate Duality

The natural conditions we introduced above imply that to design a market for a set of K securities with payoffs specified by an arbitrary payoff function $\boldsymbol{\rho} : \mathcal{O} \rightarrow \mathbb{R}_{\geq 0}^K$, we should use a cost function based market with a convex, differentiable cost function such that $\text{closure}(\{\nabla C(\mathbf{q}) : \mathbf{q} \in \mathbb{R}^K\}) = \text{closure}(\mathcal{H}(\boldsymbol{\rho}(\mathcal{O})))$. We now provide a general technique that can be used to design and compare properties of cost functions that satisfy these criteria. In order to accomplish this, we make use of tools from convex analysis, and in particular, the notion of *conjugate duality*. We begin by stating a precise definition of this notion of duality, as well as some useful results. We use the notation $\text{dom}(f)$ to refer to the domain of a function f , i.e., where it is defined and finite valued.

Definition 1 (Rockafellar [56], Section 7). *A convex function $f : \mathbb{R}^K \rightarrow [-\infty, \infty]$ is said to be closed when the epigraph of f is a closed set, or equivalently, the set $\{\mathbf{x} : f(\mathbf{x}) \leq \alpha\}$ is closed for all $\alpha \in \mathbb{R}$.*

For the remainder of the paper, we will only consider closed functions.

Definition 2 (Rockafellar [56], Section 12). *For any convex function $f : \mathbb{R}^K \rightarrow [-\infty, \infty]$, the convex conjugate f^* of f is defined as*

$$f^*(\mathbf{z}) := \sup_{\mathbf{x} \in \mathbb{R}^K} \mathbf{z} \cdot \mathbf{x} - f(\mathbf{x}).$$

The curious reader can find good discussions of conjugate functions in, e.g., Boyd and Vandenberghe [9] as well as Hiriart-Urruty and Lemaréchal [38]. We will cite two useful results from Rockafellar [56].

Theorem 3 (Rockafellar [56], Theorem 12.2 and Corollary 12.2.2). *For any closed convex function $f : \mathbb{R}^K \rightarrow [-\infty, \infty]$, the conjugate f^* is also closed and convex, and $f^{**} = f$. Furthermore, we can write*

$$f^*(\mathbf{y}) = \sup_{\mathbf{x} \in \text{relint}(\text{dom}(f))} \mathbf{y} \cdot \mathbf{x} - f(\mathbf{x}).$$

This theorem tells us two things. First, when taking the sup, we don't have to worry about what happens on the boundary. Second, it effectively tells us that there is a one-to-one correspondence between every closed convex function and its dual. How do various properties of the function translate when we go to the dual? We give one useful result, showing that *differentiability* is a dual property to *strict convexity*.

Theorem 4 (Rockafellar [56], Theorem 26.3). *Given a proper⁴ closed convex function $f : \mathbb{R}^K \rightarrow [-\infty, \infty]$, f is finite and differentiable everywhere on \mathbb{R}^K if and only if its conjugate f^* is strictly convex on $\text{dom}(f^*)$.*

We will use this notion of conjugate duality to aid in constructing cost functions C which satisfy our desired properties. Clearly, for any cost function C , we can construct its conjugate, which we will henceforth refer to as R . A more interesting and useful question to examine is under what conditions we obtain a valid cost function if we construct R first and set $C := R^*$.

Theorem 5. *Assume we have an outcome space \mathcal{O} and a payoff function ρ such that $\rho(\mathcal{O})$ is a bounded subset of \mathbb{R}^K . Then for any cost function $C : \mathbb{R}^K \rightarrow \mathbb{R}$ satisfying Conditions 2-5 and where C is closed, there exists a function $R : \mathbb{R}^K \rightarrow [-\infty, \infty]$ such that*

$$C(\mathbf{q}) = \sup_{\mathbf{x} \in \text{relint}(\mathcal{H}(\rho(\mathcal{O})))} \mathbf{x} \cdot \mathbf{q} - R(\mathbf{x}). \quad (4)$$

Furthermore, for any proper closed convex function R defined on $\text{relint}(\mathcal{H}(\rho(\mathcal{O})))$, if R is strictly convex on its domain then the cost function defined by the conjugate, $C := R^$, satisfies Conditions 2-5.*

This theorem is the key result that will guide us in designing a market pricing mechanism. This mechanism relies on constructing a cost function $C : \mathbb{R}^K \rightarrow \mathbb{R}$ that satisfies Conditions 2-5, and we are now given ingredients to achieve this: pick *any* closed strictly convex function R with domain containing $\mathcal{H}(\rho(\mathcal{O}))$, and we can set $C := R^*$.

Duality-based cost function market maker

Input: security space \mathbb{R}^K and a bounded payoff function $\rho : \mathcal{O} \rightarrow \mathbb{R}^K$

Input: convex compact price space Π (typically assumed to be $\mathcal{H}(\rho(\mathcal{O}))$)

Input: closed strictly convex and differentiable R with $\text{relint}(\Pi) \subseteq \text{dom}(R)$

Output: cost function $C : \mathbb{R}^K \rightarrow \mathbb{R}$ defined by $C(\mathbf{q}) := \sup_{\mathbf{x} \in \text{relint}(\Pi)} \mathbf{x} \cdot \mathbf{q} - R(\mathbf{x})$

Notice that in this definition, we introduce the concept of a “price space” denoted by Π . Indeed, we will typically set the price space $\Pi = \mathcal{H}(\rho(\mathcal{O}))$, as it is this case for which Theorem 5 holds. We give the more

⁴A function f is said to be *proper* if $f(\mathbf{x}) > -\infty$ for all \mathbf{x} and $f(\mathbf{x}) < +\infty$ for some \mathbf{x} .

general definition because, as we will discuss, there can be computational benefits to allowing a Π to be larger. We also require that R be differentiable which, while not strictly necessary, is a reasonable condition and eases the notation as we can now discuss the gradient $\nabla R(\mathbf{x})$.

This duality based approach to designing the market mechanism is convenient for several reasons. First, it leads to markets that are efficient to implement whenever $\mathcal{H}(\rho(\mathcal{O}))$ can be described by a polynomial number of simple constraints⁵. The difficulty with combinatorial outcome spaces is that actually enumerating the set of outcomes can be challenging or impossible. In our proposed framework we need only work with the convex hull of the payoff vectors for each outcome when represented by a low-dimensional payoff function $\rho(\cdot)$. This has significant benefits, as one often encounters convex sets which contain exponentially many vertices yet can be described by polynomially many constraints. Moreover, as the construction of C is based entirely on convex programming, we reduce the problem of automated market making to the problem of optimization for which we have a wealth of efficient algorithms. Second, this method yields simple formulas for properties of markets that help us choose the best market to run. Two of these properties, worst-case monetary loss and worst-case information loss, are analyzed below.

A note on relative interiors and boundaries. In order to establish precise statements, our discussions about certain convex sets – e.g. $\{\nabla C\}$, $\mathcal{H}(\rho(\mathcal{O}))$, and Π – have required precise definitions like the relative boundary and interior, and the closure of a set. One might ask whether this is necessary, as we might be focusing too heavily on “boundary cases.” While these details are occasionally cumbersome, they are important and do arise for very simple markets. For example, for the case of a complete market on n outcomes using the LMSR cost function $C(\mathbf{q}) = b \log \sum_i \exp(q_i/b)$, we have that $\{\nabla C(\mathbf{q}) : \mathbf{q} \in \mathbb{R}^n\} = \text{relint}(\Delta_n)$; prices of 0 and 1 can be reached only in the limit.

A note on the Legendre Transformation. Given an arbitrary smooth convex function f , we can define the *Legendre Transformation* which maps a point $\mathbf{x} \in \text{dom}(f)$ via the rule $\mathbf{x} \mapsto \nabla f(\mathbf{x})$. Indeed, under certain circumstances we get that this map is the *inverse* of the Legendre transformation of the conjugate f^* , i.e., $\nabla f^*(\nabla f(\mathbf{x})) = \mathbf{x}$ and $\nabla f(\nabla f^*(\mathbf{y})) = \mathbf{y}$ for every $\mathbf{x} \in \text{dom}(f)$ and $\mathbf{y} \in \text{dom}(f^*)$. Unfortunately the latter only holds when f is strictly convex and the interior of $\text{dom}(f)$ is non-empty (see Rockafellar, chapter 26 [56]). So while we would like to argue that ∇C is the inverse of the map ∇R for our framework, this will generally not be true. We may still state the following useful result [56]:

$$C(\mathbf{q}) = \mathbf{q} \cdot \nabla C(\mathbf{q}) - R(\nabla C(\mathbf{q})) \quad \text{for all } \mathbf{q} \in \mathbb{R}^K.$$

This fact is quite helpful when we recall that $C(\mathbf{q})$ is defined as the supremum over $\mathbf{x} \in \text{relint}(\Pi)$ of $\mathbf{q} \cdot \mathbf{x} - R(\mathbf{x})$. As long as $\nabla C(\mathbf{q})$ is contained in Π , it follows immediately that the supremum is achieved at $\mathbf{x} = \nabla C(\mathbf{q})$, and by the strict convexity of R this solution must be unique.

4.1 Bounding Market Maker Loss and Loss of Information

We now discuss two key properties of our proposed market framework. We will make use of the notion of a Bregman divergence. The *Bregman divergence* with respect to a convex function f is given by

$$D_f(\mathbf{x}, \mathbf{y}) := f(\mathbf{x}) - f(\mathbf{y}) - \nabla f(\mathbf{y})(\mathbf{x} - \mathbf{y}).$$

It is clear by convexity that $D_f(\mathbf{x}, \mathbf{y}) \geq 0$ for all \mathbf{x} and \mathbf{y} .

⁵A convex program can be solved with arbitrarily small error ϵ in time polynomial of $1/\epsilon$ and the size of the problem input using standard techniques such as the interior-point method. In this paper, we do not worry about finding the exact solution to the convex programs.

4.1.1 Bounding the Market Maker's Monetary Loss

When comparing market mechanisms, it is useful to consider the market maker's worst-case monetary loss,

$$\sup_{\mathbf{q} \in \mathbb{R}^K} \left(\sup_{\mathfrak{o} \in \mathcal{O}} (\boldsymbol{\rho}(\mathfrak{o}) \cdot \mathbf{q}) - C(\mathbf{q}) + C(\mathbf{0}) \right).$$

This quantity is simply the worst-case difference between the maximum amount that the market maker might have to pay the traders ($\sup_{\mathfrak{o} \in \mathcal{O}} \boldsymbol{\rho}(\mathfrak{o}) \cdot \mathbf{q}$) and the amount of money collected by the market maker ($C(\mathbf{q}) - C(\mathbf{0})$). The following theorem provides a bound on this loss in terms of the conjugate function.

Theorem 6. *Consider any duality-based cost function market maker with $\Pi = \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$. The worst-case monetary loss of the market maker is no more than*

$$\sup_{\mathbf{x} \in \boldsymbol{\rho}(\mathcal{O})} R(\mathbf{x}) - \min_{\mathbf{x} \in \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))} R(\mathbf{x}). \quad (5)$$

Furthermore, the above bound is tight, as the supremum of the market maker loss, over all quantity vectors \mathbf{q} and outcomes \mathfrak{o} , is exactly the value in Equation 5.

Proof. Let \mathbf{q} denote the final vector of quantities sold, $\nabla C(\mathbf{q})$ denote the final vector of instantaneous prices, and \mathfrak{o} denote the true outcome. From Equation 4, we have that $C(\mathbf{q}) = \nabla C(\mathbf{q}) \cdot \mathbf{q} - R(\nabla C(\mathbf{q}))$ and $C(\mathbf{0}) = -\min_{\mathbf{x} \in \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))} R(\mathbf{x})$. The difference between the amount that the market maker must pay out and the amount that the market maker has previously collected is then

$$\begin{aligned} & \boldsymbol{\rho}(\mathfrak{o}) \cdot \mathbf{q} - C(\mathbf{q}) + C(\mathbf{0}) \\ &= \boldsymbol{\rho}(\mathfrak{o}) \cdot \mathbf{q} - (\nabla C(\mathbf{q}) \cdot \mathbf{q} - R(\nabla C(\mathbf{q}))) - \min_{\mathbf{x} \in \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))} R(\mathbf{x}) \\ &= \mathbf{q} \cdot (\boldsymbol{\rho}(\mathfrak{o}) - \nabla C(\mathbf{q})) + R(\nabla C(\mathbf{q})) - \min_{\mathbf{x} \in \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))} R(\mathbf{x}) + R(\boldsymbol{\rho}(\mathfrak{o})) - R(\boldsymbol{\rho}(\mathfrak{o})) \\ &= R(\boldsymbol{\rho}(\mathfrak{o})) - \min_{\mathbf{x} \in \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))} R(\mathbf{x}) - (R(\boldsymbol{\rho}(\mathfrak{o})) - R(\nabla C(\mathbf{q})) - \mathbf{q} \cdot (\boldsymbol{\rho}(\mathfrak{o}) - \nabla C(\mathbf{q}))) \quad (6) \\ &\leq R(\boldsymbol{\rho}(\mathfrak{o})) - \min_{\mathbf{x} \in \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))} R(\mathbf{x}) - (R(\boldsymbol{\rho}(\mathfrak{o})) - R(\nabla C(\mathbf{q})) - \nabla R(\nabla C(\mathbf{q})) \cdot (\boldsymbol{\rho}(\mathfrak{o}) - \nabla C(\mathbf{q}))) \\ &= R(\boldsymbol{\rho}(\mathfrak{o})) - \min_{\mathbf{x} \in \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))} R(\mathbf{x}) - D_R(\boldsymbol{\rho}(\mathfrak{o}), \nabla C(\mathbf{q})), \end{aligned}$$

where D_R is the Bregman divergence with respect to R , as defined above. The inequality follows from the first-order optimality condition for convex optimization, which says that for any convex and differentiable f defined on the domain Π , if f is minimized at \mathbf{x} , then

$$\nabla f(\mathbf{x}) \cdot (\mathbf{y} - \mathbf{x}) \leq 0 \quad \text{for any } \mathbf{y} \in \Pi.$$

Consider $f(\mathbf{x}) = R(\mathbf{x}) - \mathbf{q} \cdot \mathbf{x}$. The minimum of this function occurs at $\mathbf{x} = \nabla C(\mathbf{q})$ via the duality assumption. Plugging in $\mathbf{y} = \boldsymbol{\rho}(\mathfrak{o})$ yields the inequality.

Since the divergence is always nonnegative, this is upperbounded by $R(\boldsymbol{\rho}(\mathfrak{o})) - \min_{\mathbf{x} \in \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))} R(\mathbf{x})$, which is in turn upperbounded by $\sup_{\mathbf{x} \in \boldsymbol{\rho}(\mathcal{O})} R(\mathbf{x}) - \min_{\mathbf{x} \in \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))} R(\mathbf{x})$.

Finally, we show that this loss bound is tight. First, select any $\epsilon > 0$. Choose an outcome \mathfrak{o} so that $\sup_{\mathfrak{o}' \in \mathcal{O}} R(\boldsymbol{\rho}(\mathfrak{o}')) - R(\boldsymbol{\rho}(\mathfrak{o})) < \epsilon/2$. Next, choose some \mathbf{q}' so that $D_R(\boldsymbol{\rho}(\mathfrak{o}), \nabla C(\mathbf{q}')) < \epsilon/2$. This is achievable because the space of gradients of C is assumed to span $\text{relint}(\mathcal{H}(\boldsymbol{\rho}(\mathcal{O})))$ via Theorem 2, and so we can ensure that $\nabla C(\mathbf{q}')$ is arbitrarily close to $\boldsymbol{\rho}(\mathfrak{o})$. Finally, let $\mathbf{q} := \nabla R(\nabla C(\mathbf{q}'))$, and observe that by

construction we have $\nabla C(\mathbf{q}) = \nabla C(\mathbf{q}')$. To compute the market maker's loss for this particular choice of \mathbf{q} and σ , we apply Equation 6 to obtain:

$$\begin{aligned} & R(\boldsymbol{\rho}(\sigma)) - \min_{\mathbf{x} \in \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))} R(\mathbf{x}) - (R(\boldsymbol{\rho}(\sigma)) - R(\nabla C(\mathbf{q})) - \mathbf{q} \cdot (\boldsymbol{\rho}(\sigma) - \nabla C(\mathbf{q}))) \\ &= R(\boldsymbol{\rho}(\sigma)) - \min_{\mathbf{x} \in \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))} R(\mathbf{x}) - D_R(\boldsymbol{\rho}(\sigma), \nabla C(\mathbf{q})) \\ &> \sup_{\sigma' \in \mathcal{O}} R(\boldsymbol{\rho}(\sigma')) - \min_{\mathbf{x} \in \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))} R(\mathbf{x}) - \epsilon \end{aligned}$$

where the first equality holds by the definition of the Bregman divergence, because $\mathbf{q} = \nabla R(\nabla C(\mathbf{q}))$. \square

This theorem tells us that as long as the conjugate function is bounded on $\mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$, the market maker's worst-case loss is also bounded⁶. It says further that this loss is actually realized, for a particular outcome σ , at least when the price vector approaches σ . This suggests that loss to the market maker is worst when the traders are the most certain about the outcome.

4.1.2 Bounding Information Loss

Information loss can occur when securities are sold in discrete quantities (for example, single units), as they are in most real-world markets. Without the ability to purchase arbitrarily small bundles, traders may not be able to change the market prices to reflect their true beliefs about the expected payoff of each security, even if expressiveness is satisfied. We will argue that the amount of information loss is captured by the market's bid-ask spread for the smallest trading unit. Given some \mathbf{q} , the current bid-ask spread of security bundle \mathbf{r} is defined to be $(C(\mathbf{q} + \mathbf{r}) - C(\mathbf{q})) - (C(\mathbf{q}) - C(\mathbf{q} - \mathbf{r}))$. This is simply the difference between the current cost of buying the bundle \mathbf{r} and the current price at which \mathbf{r} could be sold.

To see how the bid-ask spread relates to information loss, suppose that the current vector of quantities sold is \mathbf{q} . If securities must be sold in unit chunks, a rational, risk-neutral trader will not buy security i unless she believes the expected payoff of this security is at least $C(\mathbf{q} + \mathbf{e}_i) - C(\mathbf{q})$, where \mathbf{e}_i is the vector that has value 1 at its i th element and 0 everywhere else. Similarly, she will not sell security i unless she believes the expected payoff is at most $C(\mathbf{q}) - C(\mathbf{q} - \mathbf{e}_i)$. If her estimate of the expected payoff of the security is between these two values, she has no incentive to buy or sell the security. In this case, it is only possible to infer that the trader believes the true expected payoff lies somewhere in the range $[C(\mathbf{q}) - C(\mathbf{q} - \mathbf{e}_i), C(\mathbf{q} + \mathbf{e}_i) - C(\mathbf{q})]$. The bid-ask spread is precisely the size of this range.

The bid-ask spread depends on how fast instantaneous prices change as securities are bought or sold. Intuitively, the bid-ask spread relates to the depth of the market. When the bid-ask spread is large, new purchases or sales can change the prices of the securities dramatically; essentially, the market is shallow. When the bid-ask spread is small, purchases or sales may only move the prices slightly; the market is deep. Based on this intuition, for complete markets, Chen and Pennock [14] use the inverse of $\partial^2 C(\mathbf{q}) / \partial q_i^2$ to capture the notion of market depth for each security i independently. In a similar spirit, we define a *market depth parameter*, β , for our complex securities markets with twice-differentiable C . Larger values of β correspond to deeper markets. We will bound the bid-ask spread in terms of this parameter, and use this parameter to show that there exists a clear tradeoff between worst-case monetary loss and information loss; this will be formalized in Theorem 7 below.

⁶In Section 6, we will state a more general, stronger bound on market maker loss capturing the intuitive notion that the market maker's profits should be higher when the distance between the final vector of prices and the payoff vector $\boldsymbol{\rho}(\sigma)$ of the true outcome σ is large; see Theorem 9.

Definition 3. For any duality-based cost function market maker, if C is twice-differentiable, the market depth parameter $\beta(\mathbf{q})$ for a quantity vector \mathbf{q} is defined as $\beta(\mathbf{q}) = 1/V_c(\mathbf{q})$, where $V_c(\mathbf{q})$ is the largest eigenvalue of $\nabla^2 C(\mathbf{q})$, the Hessian of C at \mathbf{q} . The worst-case market depth is $\beta = \inf_{\mathbf{q} \in \mathbb{R}^K} \beta(\mathbf{q})$.

Let $\text{relint}(\Pi)$ be the relative interior of Π . If C is twice-differentiable, then for any \mathbf{q} such that $\nabla C(\mathbf{q}) \in \text{relint}(\Pi)$, we have a correspondence between the Hessian of C at \mathbf{q} and the Hessian of R at $\nabla C(\mathbf{q})$. More precisely, we have that $\mathbf{u}^\top \nabla^2 C(\mathbf{q}) \mathbf{u} = \mathbf{u}^\top \nabla^2 R(\nabla C(\mathbf{q})) \mathbf{u}$ for any $\mathbf{u} = \mathbf{x} - \mathbf{x}'$ with $\mathbf{x}, \mathbf{x}' \in \Pi$. (See, for example, Gorni [29] for more.) This means that $\beta(\mathbf{q})$ is equivalently defined as the smallest eigenvalue of $\nabla^2 R(\nabla C(\mathbf{q}))|_{\Pi}$; that is, where we consider the second derivative only within the price region Π .

The definition of worst-case market depth implies that $1/\beta$ is an upper bound on the curvature of C , which implies that C is locally bounded by a quadratic with Hessian I/β . We can derive the following.

Lemma 1. Consider a duality-based cost function market maker with worst-case market depth β . If C is twice differentiable, then for any \mathbf{q} and \mathbf{r} we have

$$D_C(\mathbf{q} + \mathbf{r}, \mathbf{q}) \leq \frac{\|\mathbf{r}\|^2}{2\beta}.$$

Proof. Let $f(t) := D_C(\mathbf{q} + t\mathbf{r}/\|\mathbf{r}\|, \mathbf{q})$. Notice that $f(0) = 0$, and $f'(0) = 0$ since C is differentiable and $D_C(\mathbf{x}, \mathbf{q})$ is minimized at $\mathbf{x} = \mathbf{q}$ or, equivalently, $f(t)$ is minimized at $t = 0$. It follows that $D_C(\mathbf{q} + \mathbf{r}, \mathbf{q}) = \int_0^{\|\mathbf{r}\|} \int_0^s f''(t) dt ds$ by standard calculus arguments. However, $f''(t)$ is always smaller than the largest eigenvalue of $\nabla^2 C(\mathbf{q} + t\mathbf{r}/\|\mathbf{r}\|)$ which, by definition, is always smaller than $1/\beta$. It follows that

$$D_C(\mathbf{q} + \mathbf{r}, \mathbf{q}) = \int_0^{\|\mathbf{r}\|} \int_0^s f''(t) dt ds \leq \int_0^{\|\mathbf{r}\|} \int_0^s \frac{1}{\beta} dt ds = \int_0^{\|\mathbf{r}\|} \frac{s}{\beta} ds = \frac{\|\mathbf{r}\|^2}{2\beta}$$

as desired. \square

It is easy to verify that the bid-ask spread can be written in terms of Bregman divergences. In particular, $C(\mathbf{q} + \mathbf{r}) - C(\mathbf{q}) - (C(\mathbf{q}) - C(\mathbf{q} - \mathbf{r})) = D_C(\mathbf{q} + \mathbf{r}, \mathbf{q}) + D_C(\mathbf{q} - \mathbf{r}, \mathbf{q})$. This implies that the worst-case bid-ask spread of a market with market depth β can be upperbounded by a constant times $1/\beta$. That is, as the market depth parameter increases, the bid-ask spread must decrease. The following theorem shows that this leads to an inherent tension between worst-case monetary loss and information loss. Here $\text{diam}(\mathcal{H}(\rho(\mathcal{O})))$ denotes the diameter of the hull of the payoff vectors for each outcome.

Theorem 7. For any duality-based cost function market maker with worst-case market depth β , for any \mathbf{r}, \mathbf{q} meeting the conditions in Lemma 1, the bid-ask spread for bundle \mathbf{r} with previous purchases \mathbf{q} is no more than $\|\mathbf{r}\|^2/\beta$. The worst-case monetary loss of the market maker is at least $\beta \cdot \text{diam}^2(\mathcal{H}(\rho(\mathcal{O}))) / 8$.

Proof. The bound on the bid-ask spread follows immediately from Lemma 1 and the argument above. The value β lower-bounds the eigenvalues of R everywhere on Π . Hence, if we do a quadratic lower-bound of R from the point $\mathbf{x}_0 = \arg \min_{\mathbf{x} \in \Pi} R(\mathbf{x})$ with Hessian defined by βI , then we see that $R(\mathbf{x}) - R(\mathbf{x}_0) \geq D_R(\mathbf{x}, \mathbf{x}_0) \geq \frac{\beta}{2} \|\mathbf{x} - \mathbf{x}_0\|^2$. In the worst-case, $\|\mathbf{x} - \mathbf{x}_0\| = \text{diam}(\mathcal{H}(\rho(\mathcal{O}))) / 2$, which finishes the proof. \square

We can see that there is a direct tradeoff between the upper bound⁷ of the bid-ask spread, which shrinks as β grows, and the lower bound of the worst-case loss of the market maker, which grows linearly in β . This

⁷Strictly speaking, as we are emphasizing the necessary tradeoff between bid-ask spread and worst-case loss, we should have a *lower bound* on the bid-ask spread. On the other hand, if the worst-case market depth parameter is β then there is some \mathbf{q} and \mathbf{r} such that $D_C(\mathbf{q} + \mathbf{r}, \mathbf{q})/\|\mathbf{r}\|^2 \approx 1/(2\beta)$ and this approximation can be made arbitrarily tight for small enough \mathbf{r} when C is twice differentiable.

tradeoff is very intuitive. When the market is shallow (small β), small trades have a large impact on market prices, and traders cannot purchase too many shares of the same security without paying a lot. When the market is deep (large β), prices change slowly, allowing the market maker to gain more precise information, but simultaneously forcing the market maker to take on more risk since many shares of a security can be purchased at prices that are potentially too low. This tradeoff can be adjusted by scaling R , which scales β . This is analogous to adjusting the “liquidity parameter” b of the LMSR.

4.2 Selecting a Conjugate Function

We have seen that the choice of the conjugate function R impacts market properties such as worst-case loss and information loss. We now explore this choice in more detail.

In many situations, the ideal choice of the conjugate is a function of the form

$$R(\mathbf{x}) := \frac{\lambda}{2} \|\mathbf{x} - \mathbf{x}_0\|^2. \quad (7)$$

Here $R(\mathbf{x})$ is simply the squared Euclidean distance between \mathbf{x} and an initial price vector $\mathbf{x}_0 \in \Pi$, scaled by $\lambda/2$. By utilizing this quadratic conjugate function, we achieve a market depth that is uniformly λ over the entire security space. Furthermore, if \mathbf{x}_0 is chosen as the “center” of Π , namely $\mathbf{x}_0 = \arg \min_{\mathbf{x} \in \Pi} \max_{\mathbf{y} \in \Pi} \|\mathbf{x} - \mathbf{y}\|$, then the worst-case loss of the market maker is $\max_{\mathbf{x} \in \Pi} R(\mathbf{x}) = (\lambda/8) \text{diam}^2(\Pi)$. While the market maker can tune λ appropriately according to the desired tradeoff between worst-case market depth and worst-case loss, the tradeoff is tightest when R has a Hessian that is uniformly a scaled identity matrix, or more precisely where R takes the form in Equation 7.

Unfortunately, by selecting a conjugate of this form, or any R with bounded derivative, the market maker does inherit one potentially undesirable property: security prices may become constant when $\nabla C(\mathbf{q})$ reaches a point at $\text{relbnd}(\Pi)$, the relative boundary of Π (see Section 4.1). That is, if we arrive at a total demand \mathbf{q} where $\nabla C(\mathbf{q}) = \boldsymbol{\rho}(\sigma)$ for some outcome σ , our mechanism begins offering securities at a price *equal to the best-case payoff*, akin to asking someone to bet a dollar for the chance to possibly win a dollar. The Quad-SCPM for complete markets is known to exhibit this behavior [4].

To avoid these undesirable pricing scenarios, it is sufficient to require that our conjugate function satisfies one condition. We say that a convex function R defined on Π is a *pseudo-barrier*⁸ for Π if $\|\nabla R(\mathbf{x}_t)\| \rightarrow \infty$ for any sequence of points $\mathbf{x}_1, \mathbf{x}_2, \dots \in \Pi$ which tends towards $\text{relbnd}(\Pi)$. If we require our conjugate function R to be a pseudo-barrier, we are guaranteed that the instantaneous price vector $\nabla C(\mathbf{q})$ always lies in $\text{relint}(\Pi)$, and does not become constant near the boundary.

It is important to note that, while it is desirable that $\|\nabla R(\mathbf{x}_t)\| \rightarrow \infty$ as \mathbf{x}_t approaches $\text{relbnd}(\Pi)$, it is generally not desirable that $R(\mathbf{x}_t) \rightarrow \infty$. Recall that the market maker’s worst-case loss grows with the maximum value of R on Π and thus we should hope for a conjugate function that is bounded on the domain. A perfect example of convex function that is simultaneously bounded and a pseudo-barrier is the *negative entropy function* $H(\mathbf{x}) = \sum_i x_i \log x_i$, defined on the n -simplex Δ_n . It is perhaps no surprise that LMSR, the most common market mechanism for complete security spaces, can be described by the choice $R(\mathbf{x}) := bH(\mathbf{x})$ where the price space $\Pi = \Delta_n$ [4, 15].

⁸We use the term pseudo-barrier to distinguish this from the typical definition of a barrier function on a set Π , which is a function that grows without bound towards the boundary of Π . The term *Legendre* was introduced by Cesa-Bianchi and Lugosi [13] for a similar notion, yet their definition requires the stronger condition that Π contain a nonempty interior.

5 Examples of Computationally Efficient Markets

In the previous section, we provided a general framework for designing markets on combinatorial or infinite outcome spaces. We now provide some examples of markets that can be operated efficiently using this framework.

5.1 Subset Betting

Recall the scenario described in Section 3.1 in which the outcome is a ranking of a set of n competitors, such as n horses in a race, represented as a permutation $\pi : [n] \rightarrow [n]$. Chen et al. [17] proposed a betting language, *subset betting* in which traders can place bets (i, j) , for any candidate i and any slot j , that pay out \$1 in the event that $\pi(i) = j$ and \$0 otherwise.⁹ Chen et al. [18] showed that pricing bets of this form using LMSR is #P-hard and provided an algorithm for approximating the prices by exploiting the structure of the market. Using our framework, it is simple to design a computationally efficient market for securities of this form.

In order to set up such a combinatorial market within our framework, we must be able to efficiently work with the convex hull of the payoff vectors for each outcome. Notice that, for an outcome π , the associated payoff can be described by a matrix M_π , with $M_\pi(i, j) = I[\pi(i) = j]$, where $I[\cdot]$ is the indicator function. Taking this one step further, it is easy to verify that the convex hull of the set of permutation matrices is precisely the set of *doubly stochastic matrices*, that is the set

$$\Pi = \left\{ X \in \mathbb{R}_{\geq 0}^{n \times n} : \sum_{i'=1}^n X(i', j) = \sum_{j'=1}^n X(i, j') = 1 \quad \forall i, j \right\},$$

where $X(i, j)$ represents the element at the i th row and j th column of the matrix X . Notice, importantly, that this set is described by only n^2 variables and $O(n)$ constraints.

To fully specify the market maker, we must also select a conjugate function R for our price space. While the quadratic conjugate function is an option, there is a natural extension of the entropy function, whose desirable properties were discussed in the previous section, for the space of stochastic matrices. For any $X \in \Pi$, let us set

$$R(X) = \lambda \sum_{i,j} X(i, j) \log X(i, j).$$

The worst-case market depth is computed as the minimum of the smallest eigenvalue of the Hessian of R within the $\text{relint}(\Pi)$. This occurs at the matrix with all values $1/n$, hence the worst-case depth is $n\lambda$. The worst-case loss, on the other hand, is easily computed as $\lambda n \log n$.

5.2 Sphere Betting

One important challenge of operating a combinatorial prediction market is to always maintain the logical consistency of security prices. Our framework offers a way to incorporate the constraints on security prices into pricing. Hence, in addition to combinatorial prediction markets, our framework can be used to design markets where security prices have some natural constraints due to their problem domains.

⁹The original definition of subset betting allowed bets of the form “any candidate in set S will end up in slot j ” or “candidate i will end up in one of the slots in set S .” A bet of this form can be constructed easily using our betting language by bundling multiple securities.

We consider an example in which the outcome space is infinite. An object orbiting the planet, perhaps a satellite, is predicted to fall to earth in the near future and will land at an unknown location, which we would like to predict. We represent locations on the earth as unit vectors $\mathbf{u} \in \mathbb{R}^3$. The difficulty of this example arises from the fact that the outcome must be a unit vector, imposing constraints on the three coordinates. We will design a market with three securities, each corresponding to one coordinate of the final location of the object. In particular, security i will pay off $u_i + 1$ dollars if the object lands in location \mathbf{u} . (The addition of 1, while not strictly necessary, ensures that the payoffs, and therefore prices, remain positive, though it will be necessary for traders to sell securities to express certain beliefs.) This means that traders can purchase security bundles $\mathbf{r} \in \mathbb{R}^3$ and, when the object lands at a location \mathbf{u} , receive a payoff $(\mathbf{u} + \mathbf{1}) \cdot \mathbf{r}$. Note that in this example, the outcome space is infinite, but the security space is small.

The price space $\mathcal{H}(\rho(\mathcal{O}))$ for this market will be the 2-norm unit ball centered at $\mathbf{1}$. To construct a market for this scenario, let us make the simple choice of $R(\mathbf{x}) = \lambda \|\mathbf{x} - \mathbf{1}\|^2$ for some parameter $\lambda > 0$. When $\|\mathbf{q}\| \leq 2\lambda$, there exists an \mathbf{x} such that $\nabla R(\mathbf{x}) = \mathbf{q}$. In particular, this is true for $\mathbf{x} = (1/2)\mathbf{q}/\lambda + \mathbf{1}$, and $\mathbf{q} \cdot \mathbf{x} - R(\mathbf{x})$ is minimized at this point. When $\|\mathbf{q}\| > 2\lambda$, $\mathbf{q} \cdot \mathbf{x} - R(\mathbf{x})$ is minimized at an \mathbf{x} on the boundary of $\mathcal{H}(\rho(\mathcal{O}))$. Specifically, it is minimized at $\mathbf{x} = \mathbf{q}/\|\mathbf{q}\| + \mathbf{1}$. From this, we can compute

$$C(\mathbf{q}) = \begin{cases} \frac{1}{4\lambda} \|\mathbf{q}\|^2 + \mathbf{q} \cdot \mathbf{1}, & \text{when } \|\mathbf{q}\| \leq 2\lambda, \\ \|\mathbf{q}\| + \mathbf{q} \cdot \mathbf{1} - \lambda, & \text{when } \|\mathbf{q}\| > 2\lambda. \end{cases}$$

The market depth parameter β is 2λ ; in fact, $\beta(\mathbf{x}) = 2\lambda$ for any price vector \mathbf{x} in the interior of $\mathcal{H}(\rho(\mathcal{O}))$. By Theorem 6, the worst-case loss of the market maker is no more than λ , which is precisely the lower bound implied by Theorem 7. Finally, the divergence $D_C(\mathbf{q} + \mathbf{r}, \mathbf{q}) \leq \|\mathbf{r}\|^2/(4\lambda)$ for all \mathbf{q}, \mathbf{r} , with equality when $\|\mathbf{q}\|, \|\mathbf{q} + \mathbf{r}\| \leq 2\lambda$, implying that the bid-ask spread scales linearly with $\|\mathbf{r}\|^2/\lambda$.

We note that for this particular prediction problem, if we try to predict the latitude and longitude of the landing location, we don't have any constraints on prices. In particular, we can have two securities that pay off linearly with the latitude and longitude of the landing location respectively. These two securities are independent and can be traded in two independent markets.

6 Computational Complexity and Relaxations

In Section 3, we argued that the space of feasible price vectors should be precisely $\mathcal{H}(\rho(\mathcal{O}))$, the convex hull of the payoff vectors for each outcome. In each of our examples, we have discussed market scenarios for which this hull has a polynomial number of constraints, allowing us to efficiently set prices via convex optimization. Unfortunately, one should not necessarily expect that a given payoff function and outcome space will lead to an efficiently describable convex hull. In this section, we explore a couple of approaches to overcome such complexity challenges. First, we discuss the case in which $\mathcal{H}(\rho(\mathcal{O}))$ has exponentially (or infinitely) many constraints yet gives rise to a *separation oracle*. Second, we show that the price space Π can indeed be *relaxed* beyond $\mathcal{H}(\rho(\mathcal{O}))$ without increasing the risk to the market maker. Finally, we show how this relaxation applies in practice.

6.1 Separation Oracles

If we encounter a convex hull $\mathcal{H}(\rho(\mathcal{O}))$ with exponentially-many constraints, all may not be lost. Recall that, in order to set prices, we need to solve the optimization problem $\max_{\mathbf{x} \in \mathcal{H}(\rho(\mathcal{O}))} \mathbf{q} \cdot \mathbf{x} - R(\mathbf{x})$. Under certain circumstances this can still be solved efficiently.

Consider a convex optimization problem with a concave objective function $f(\mathbf{x})$ and constraints $g_i(\mathbf{x}) \leq 0$ for all i in some index set I . That is, we want to solve:

$$\begin{aligned} \max \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{x} \in \mathbb{R}^d \\ & g_i(\mathbf{x}) \leq 0 \quad \forall i \in I \end{aligned}$$

This can be converted to a problem with a linear objective in the standard way:

$$\begin{aligned} \max \quad & c \\ \text{s.t.} \quad & \mathbf{x} \in \mathbb{R}^d, c \in \mathbb{R} \\ & f(\mathbf{x}) \geq c \\ & g_i(\mathbf{x}) \leq 0 \quad \forall i \in I \end{aligned}$$

Of course, if I is an exponentially or infinitely large set we will have trouble solving this problem directly. On the other hand, the constraint set may admit an efficient *separation oracle*, defined as a function that takes as input a point (\mathbf{x}, c) and returns `true` if all the necessary constraints are satisfied or, otherwise, returns `false` and specifies a violated constraint¹⁰. Given an efficient separation oracle one has access to alternative methods for optimization, the most famous being Khachiyan’s *ellipsoid method*, that run in polynomial time. For more details see, for example, Grötschel et al [30].

This suggests that a fruitful direction for designing computationally efficient market makers is to examine the pricing problem on an instance-by-instance basis, and for a particular instance of interest, leverage the structure of the instance to develop an efficient algorithm for solving the specific separation problem. We leave this for future research.

6.2 Relaxation of the Price Space

When dealing with a convex hull $\mathcal{H}(\rho(\mathcal{O}))$ that has a prohibitively large constraint set and does not admit an efficient separation oracle we still have one tool at our disposal: we can modify $\mathcal{H}(\rho(\mathcal{O}))$ to get an alternate price space Π which we *can* work with efficiently. Recall that in Section 3, we arrived at the requirement that $\Pi = \mathcal{H}(\rho(\mathcal{O}))$ as a necessary conclusion of the proposed conditions on our market maker. If we wish to violate this requirement, we need to consider which conditions must be weakened and revise the resulting guarantees from Section 3.

We will continue to construct our markets in the usual way, via the tuple $(\mathcal{O}, \rho, \Pi, R)$ where \mathcal{O} is the outcome space, ρ is the payoff function, $\Pi \subseteq \mathbb{R}^d$ is a convex compact set of feasible prices, and $R : \mathbb{R}^d \rightarrow \mathbb{R}$ is a strictly convex function with domain Π . The market’s cost function C will be the conjugate of R with respect to the set Π , as in Equation 4. The only difference is that we now allow Π to be distinct from $\mathcal{H}(\rho(\mathcal{O}))$. Not surprisingly, the choice of Π will affect the interest of the traders and the market maker. We prove several claims which will aid us in our market design. Theorem 8 tells us that the expressiveness condition should not be relaxed, while Theorem 9 tells us that the no-arbitrage condition can be. Together, these imply that we may safely choose Π to be a *superset* of $\mathcal{H}(\rho(\mathcal{O}))$.

Theorem 8. *For any duality-based cost function market maker, the worst-case loss of the market maker is unbounded if $\rho(\mathcal{O}) \notin \Pi$.*

¹⁰More precisely, a separation oracle returns any *separating hyperplane* that divides the input from the feasible set.

Proof. Consider some outcome \mathfrak{o} such that $\rho(\mathfrak{o}) \notin \Pi$. The feasible price set $\Pi = \{\nabla C(\mathbf{q}) : \forall \mathbf{q}\}$ is compact. Because $\rho(\mathfrak{o}) \notin \Pi$, there exists a hyperplane that *strongly* separates Π and $\rho(\mathfrak{o})$. In other words, there exists an $k > 0$ such that $\|\rho(\mathfrak{o}) - \nabla C(\mathbf{q})\| \geq k, \forall \mathbf{q}$.

When outcome \mathfrak{o} is realized, $B(\mathbf{q}) = \rho(\mathfrak{o}) \cdot \mathbf{q} - C(\mathbf{q}) + C(\mathbf{0})$ is the market maker's loss given \mathbf{q} . We have $\nabla B(\mathbf{q}) = \rho(\mathfrak{o}) - \nabla C(\mathbf{q})$, which represents the instantaneous change of the market maker's loss. For infinitesimal ϵ , let $\mathbf{q}' = \mathbf{q} + \epsilon(\rho(\mathfrak{o}) - \nabla C(\mathbf{q}))$. Then

$$\begin{aligned} B(\mathbf{q}') &= B(\mathbf{q}) + \nabla B(\mathbf{q}) \cdot [\epsilon(\rho(\mathfrak{o}) - \nabla C(\mathbf{q}))] \\ &= B(\mathbf{q}) + \epsilon \|\rho(\mathfrak{o}) - \nabla C(\mathbf{q})\|^2 \leq B(\mathbf{q}) + \epsilon k^2. \end{aligned}$$

This shows that for any \mathbf{q} we can find a \mathbf{q}' such that the market maker's worst-case loss is at least increased by ϵk^2 . This process can continue for infinite steps. Hence, we conclude that the market maker's loss is unbounded. \square

This (perhaps surprising) theorem tells us that expressiveness is not only useful for information aggregation, it is actually *necessary* for the market maker to avoid unbounded loss. The proof involves showing that if \mathfrak{o} is the final outcome and $\rho(\mathfrak{o}) \notin \Pi$, then it is possible to make an infinite sequence of trades such that each trade causes a constant amount of loss to the market maker.

In the following theorem, which is a simple extension of Theorem 6, we see that including additional price vectors in Π does not adversely impact the market maker's worst-case loss, despite the fact that the no-arbitrage condition is violated.

Theorem 9. *Consider any duality-based cost function market maker with R and Π satisfying $\sup_{\mathbf{x} \in \mathcal{H}(\rho(\mathcal{O}))} R(\mathbf{x}) < \infty$ and $\mathcal{H}(\rho(\mathcal{O})) \subseteq \Pi$. Assume that the initial price vector satisfies $\nabla C(\mathbf{0}) \in \mathcal{H}(\rho(\mathcal{O}))$. Let \mathbf{q} denote the vector of quantities sold and \mathfrak{o} denote the true outcome. The monetary loss of the market maker is no more than*

$$R(\rho(\mathfrak{o})) - \min_{\mathbf{x} \in \mathcal{H}(\rho(\mathcal{O}))} R(\mathbf{x}) - D_R(\rho(\mathfrak{o}), \nabla C(\mathbf{q})).$$

Proof. This proof is nearly identical to the proof of Theorem 6. The only major difference is that now $C(\mathbf{0}) = -\min_{\mathbf{x} \in \Pi} R(\mathbf{x})$ instead of $C(\mathbf{0}) = -\min_{\mathbf{x} \in \mathcal{H}(\rho(\mathcal{O}))} R(\mathbf{x})$, but this is equivalent since we have assumed that $\nabla C(\mathbf{0}) \in \mathcal{H}(\rho(\mathcal{O}))$. $R(\rho(\mathfrak{o}))$ is still well-defined and finite since we have assumed that $\mathcal{H}(\rho(\mathcal{O})) \subseteq \Pi$. \square

This tells us that expanding Π can only help the market maker; increasing the range of $\nabla C(\mathbf{q})$ can only increase the divergence term. This may seem somewhat counterintuitive. We originally required that $\Pi \subseteq \mathcal{H}(\rho(\mathcal{O}))$ as a consequence of the no-arbitrage condition, and by relaxing this condition, we are providing traders with potential arbitrage opportunities. However, these arbitrage opportunities do not hurt the market maker. As long as the initial price vector lies in $\mathcal{H}(\rho(\mathcal{O}))$, any such situations where a trader can earn a guaranteed profit are effectively created (and paid for) by other traders! In fact, if the final price vector $\nabla C(\mathbf{q})$ falls outside the convex hull, the divergence term will be strictly positive, improving the bound.

To elaborate on this point, let's consider an example where Π is strictly larger than $\mathcal{H}(\rho(\mathcal{O}))$. Let \mathbf{q} be the current vector of purchases, and assume the associated price vector $\mathbf{x} = \nabla C(\mathbf{q})$ lies in the interior of $\mathcal{H}(\rho(\mathcal{O}))$. Consider a trader who purchases a bundle \mathbf{r} such that the new price vector leaves this set, i.e., $\mathbf{y} := \nabla C(\mathbf{q} + \mathbf{r}) \notin \mathcal{H}(\rho(\mathcal{O}))$. We claim that this choice can be strictly improved in the sense that there is an alternative bundle \mathbf{r}' whose associated profit, for any outcome \mathfrak{o} , is strictly greater than the profit for \mathbf{r} .

For simplicity, assume \mathbf{y} is an interior point of $\Pi \setminus \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$ so that $\mathbf{q} + \mathbf{r} = \nabla R(\mathbf{y})$. Define $\pi(\mathbf{y}) := \arg \min_{\mathbf{y}' \in \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))} D_R(\mathbf{y}', \mathbf{y})$, the minimum divergence projection of \mathbf{y} into $\mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$. The alternative bundle we consider is $\mathbf{r}' = \nabla R(\pi(\mathbf{y})) - \mathbf{q}$. Our trader pays $C(\mathbf{q} + \mathbf{r}) - C(\mathbf{q} + \mathbf{r}')$ less to purchase \mathbf{r}' than to purchase \mathbf{r} . Hence, for any outcome \mathfrak{o} , we see that the increased profit for \mathbf{r}' over \mathbf{r} is

$$\begin{aligned} \boldsymbol{\rho}(\mathfrak{o}) \cdot (\mathbf{r}' - \mathbf{r}) - C(\mathbf{q} + \mathbf{r}') + C(\mathbf{q} + \mathbf{r}) &> \boldsymbol{\rho}(\mathfrak{o}) \cdot (\mathbf{r}' - \mathbf{r}) + \nabla C(\mathbf{q} + \mathbf{r}') \cdot (\mathbf{r} - \mathbf{r}') \\ &= (\boldsymbol{\rho}(\mathfrak{o}) - \pi(\mathbf{y})) \cdot (\mathbf{r}' - \mathbf{r}). \end{aligned} \quad (8)$$

Notice that we achieve strict inequality precisely because $\nabla C(\mathbf{q} + \mathbf{r}') = \pi(\mathbf{y}) \neq \mathbf{y} = \nabla C(\mathbf{q} + \mathbf{r})$. Now use the optimality condition for $\pi(\mathbf{y})$ to see that, since $\boldsymbol{\rho}(\mathfrak{o}) \in \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$, $\nabla_{\pi(\mathbf{y})}(D_R(\pi(\mathbf{y}), \mathbf{y})) \cdot (\boldsymbol{\rho}(\mathfrak{o}) - \pi(\mathbf{y})) \geq 0$. It is easy to check that $\nabla_{\pi(\mathbf{y})}(D_R(\pi(\mathbf{y}), \mathbf{y})) = \nabla R(\pi(\mathbf{y})) - \nabla R(\mathbf{y}) = \mathbf{r}' - \mathbf{r}$. Combining this last expression with the inequality above and (8) tells us that the profit increase is strictly greater than $(\boldsymbol{\rho}(\mathfrak{o}) - \pi(\mathbf{y})) \cdot (\mathbf{r}' - \mathbf{r}) \geq 0$. Simply put, the trader receives a guaranteed positive increase in profit for any outcome \mathfrak{o} .

The next theorem shows that any time the price vector lies outside of $\boldsymbol{\rho}(\mathfrak{o})$, traders could profit by moving it back inside. The proof uses a nice application of minimax duality for convex-concave functions.

Theorem 10. *For any duality-based cost function market maker, given a current quantity vector \mathbf{q}_0 with current price vector $\nabla C(\mathbf{q}_0) = \mathbf{x}_0$, a trader has the opportunity to earn a guaranteed profit of at least $\min_{\mathbf{x} \in \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))} D_R(\mathbf{x}, \mathbf{x}_0)$.*

Proof. A trader looking to earn a guaranteed profit when the current quantity is \mathbf{q}_0 hopes to purchase a bundle \mathbf{r} so that the worst-case profit $\min_{\mathfrak{o} \in \mathcal{O}} \boldsymbol{\rho}(\mathfrak{o}) \cdot \mathbf{r} - C(\mathbf{q}_0 + \mathbf{r}) + C(\mathbf{q}_0)$ is as large as possible. Notice that this quantity is strictly positive since $\mathbf{r} = \mathbf{0}$, which always has 0 profit, is one option. Thus, a trader would like to solve the following objective:

$$\begin{aligned} &\max_{\mathbf{r} \in \mathbb{R}^K} \min_{\mathfrak{o} \in \mathcal{O}} \boldsymbol{\rho}(\mathfrak{o}) \cdot \mathbf{r} - C(\mathbf{q}_0 + \mathbf{r}) + C(\mathbf{q}_0) \\ &= \min_{\mathbf{x} \in \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))} \max_{\mathbf{r} \in \mathbb{R}^K} \mathbf{x} \cdot \mathbf{r} - C(\mathbf{q}_0 + \mathbf{r}) + C(\mathbf{q}_0) \\ &= \min_{\mathbf{x} \in \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))} \max_{\mathbf{r} \in \mathbb{R}^K} \mathbf{x} \cdot (\mathbf{q}_0 + \mathbf{r}) - C(\mathbf{q}_0 + \mathbf{r}) + C(\mathbf{q}_0) - \mathbf{x} \cdot \mathbf{q}_0 \\ &= \min_{\mathbf{x} \in \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))} R(\mathbf{x}) + C(\mathbf{q}_0) - \mathbf{x} \cdot \mathbf{q}_0 \\ &= \min_{\mathbf{x} \in \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))} R(\mathbf{x}) + \mathbf{x}_0 \cdot \mathbf{q}_0 - R(\mathbf{x}_0) - \mathbf{x} \cdot \mathbf{q}_0 \\ &\geq \min_{\mathbf{x} \in \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))} D_R(\mathbf{x}, \mathbf{x}_0). \end{aligned}$$

The first equality with the min / max swap holds via Sion's Minimax Theorem [59]. The last inequality was obtained using the first-order optimality condition of the solution $\mathbf{x}_0 = \arg \max_{\mathbf{x} \in \Pi} \mathbf{x} \cdot \mathbf{q}_0 - R(\mathbf{x})$ for the vector $\mathbf{x} - \mathbf{x}_0$ which holds since $\mathbf{x} \in \Pi$. \square

When $\mathbf{x}_0 \in \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$, $D_R(\mathbf{x}, \mathbf{x}_0)$ is minimized when $\mathbf{x} = \mathbf{x}_0$ and the bound is vacuous, as we would expect. The more interesting case occurs when the prices have fallen outside of $\mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$, in which case a trader is guaranteed a riskless profit by moving $\nabla C(\mathbf{q})$ to the closest point in $\mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$.

6.3 Pair Betting via Relaxation

We return our attention to the scenario where the outcome is a ranking of n competitors, as described in Section 3.1. Consider a complex market in which traders make arbitrary pair bets: for every i, j , a trader

can purchase a security which pays out \$1 whenever $\pi(i) < \pi(j)$. Like subset bets, pricing pair bets using LMSR is known to be #P-hard [18].

We can represent the payoff structure of any such outcome π by a matrix M_π defined by

$$M_\pi(i, j) = \begin{cases} 1, & \text{if } \pi(i) < \pi(j) \\ \frac{1}{2}, & \text{if } i = j \\ 0, & \text{if } \pi(i) > \pi(j). \end{cases}$$

We would like to choose our feasible price region as the set $\mathcal{H}(\{M_\pi : \pi \in S_n\})$, where S_n is the set of permutations on $[n]$. Unfortunately, the computation of this convex hull is necessarily hard: if given only a separation oracle for the set $\mathcal{H}(\{M_\pi : \pi \in S_n\})$, we could construct a linear program to solve the “minimum feedback arc set” problem, which is known to be NP-hard.

On the positive side, we see from the previous section that the market maker can work in a larger feasible price space without risking a larger loss. We thus relax our feasible price region Π to the set of matrices $X \in \mathbb{R}^{n^2}$ satisfying the intuitive set of constraints described in Section 3.1:

$$\begin{aligned} X(i, j) &\geq 0 && \forall i, j \in [n] \\ X(i, j) &= 1 - X(j, i) && \forall i, j \in [n] \\ X(i, j) + X(j, k) + X(k, i) &\geq 1 && \forall i, j, k \in [n] \end{aligned}$$

This relaxation was first discussed by Meggido [47], who referred to such matrices as *generalized order matrices*. He proved that, for $n \leq 4$, we do have $\Pi = \mathcal{H}(\{M_\pi : \pi \in S_n\})$, but gave a counterexample showing strict containment for $n = 13$. By using this relaxed price space, the market maker allows traders to bring the price vector outside of the convex hull, yet includes a set of basic (and natural) constraints on the prices. Such a market could be implemented with any strongly convex conjugate function (e.g., quadratic).

Notice that in this example, it is computationally hard in general for a trader to determine whether or not a particular price vector falls within the convex hull; if this were not the case, then we would be able to construct a separation oracle, and could price pair bets efficiently without the relaxation. Therefore, although arbitrage opportunities may be created, it is generally intractable for traders to find and exploit these opportunities.

7 Relation to Online Learning

In this section, we use our framework to explore the striking mathematical connections that exist between automated market makers and the class of Follow the Regularized Leader algorithms for online learning. While the problem of learning in an online environment appears quite different semantically from the problem of pricing securities in a prediction market, we show that the two frameworks have a strong syntactic correspondence. We begin with a brief overview of no-regret learning and the online linear optimization problem.

7.1 Online Learning and Regret-Minimizing Algorithms

Perhaps the most canonical example of online, no-regret learning is the problem of *learning from expert advice*. In the expert setting, we imagine an algorithm that must make a sequence of predictions based on the advice of a set of N *experts* and receive a corresponding sequence of *losses*.¹¹ The goal of the algorithm

¹¹This framework could be formalized equally well in terms of rewards, but losses are more common in the literature.

is to achieve a *cumulative loss* that is “almost as low” as the cumulative loss of the best performing expert in hindsight. No statistical assumptions are made about these losses. Indeed, algorithms are expected to perform well even if the sequence of losses is chosen by an adversary.

Formally, at every time step $t \in \{1, \dots, T\}$, every expert $i \in \{1, \dots, N\}$ receives a loss $\ell_{i,t} \in [0, 1]$. The cumulative loss of expert i at time T is then defined as $L_{i,T} = \sum_{t=1}^T \ell_{i,t}$. An algorithm \mathcal{A} maintains a weight $w_{i,t}$ for each expert i at time t , where $\sum_{i=1}^N w_{i,t} = 1$. These weights can be viewed as a distribution over the experts. The algorithm then receives its own instantaneous loss $\ell_{\mathcal{A},t} = \sum_{i=1}^N w_{i,t} \ell_{i,t}$, which can be interpreted as the expected loss the algorithm would receive if it always chose an expert to follow according to the current distribution. The cumulative loss of \mathcal{A} up to time T is defined in the natural way as $L_{\mathcal{A},T} = \sum_{t=1}^T \ell_{\mathcal{A},t} = \sum_{t=1}^T \sum_{i=1}^N w_{i,t} \ell_{i,t}$. Below we use the symbols ℓ_t , \mathbf{L}_t , and \mathbf{w}_t to refer to the vector of losses, vector of cumulative loss, and the vector of weights, respectively, for each expert on round t .

It is unreasonable to expect the algorithm to achieve a small cumulative loss if none of the experts perform well. For this reason, it is typical to measure the performance of an algorithm in terms of its *regret*, defined to be the difference between the cumulative loss of the algorithm and the loss of the best performing expert, that is,

$$L_{\mathcal{A},T} - \min_{i \in \{1, \dots, N\}} L_{i,T}.$$

An algorithm is said to have *no regret* if the average per time step regret approaches 0 as T approaches infinity.

The popular Randomized Weighted Majority (WM) algorithm [23, 43] is an example of a no-regret algorithm. Weighted Majority uses weights

$$w_{i,t} = \frac{e^{-\eta L_{i,t-1}}}{\sum_{j=1}^N e^{-\eta L_{j,t-1}}}, \quad (9)$$

where $\eta > 0$ is a tunable parameter known as the *learning rate*. It is well known that the regret of WM after T trials can be bounded as

$$L_{WM(\eta),T} - \min_{i \in \{1, \dots, N\}} L_{i,T} \leq \eta T + \frac{\log N}{\eta}.$$

When T is known in advance, setting $\eta = \sqrt{\log N/T}$ yields the standard $O(\sqrt{T \log N})$ regret bound.

It has been shown that the weights chosen by Weighted Majority are precisely those that minimize a combination of empirical loss and an entropic regularization term [37, 39, 40]. More specifically, the weight vector \mathbf{w}_t at time t is precisely the solution to the following minimization problem:

$$\min_{\mathbf{w} \in \Delta_N} \mathbf{w} \cdot \mathbf{L}_{t-1} - \frac{1}{\eta} \mathbf{H}(\mathbf{w})$$

where \mathbf{H} is the entropy function, $\mathbf{H}(\mathbf{w}) := -\sum_{i=1}^N w_i \log w_i$. Indeed, Weighted Majority is an example of broader class of algorithms collectively known as *Follow the Regularized Leader* (FTRL) algorithms [34, 35, 58]. The FTRL template can be applied to a wide class of learning problems that fall under a general framework commonly known as *online convex optimization* [62]. Other problems that fall into this framework include online linear pattern classification [39], online Gaussian density estimation [7], and online portfolio selection [20]. In Algorithm 1, we present a version of FTRL tailored to the *online linear optimization* problem, an extension of the expert setting in which weights \mathbf{w}_t are chosen from a fixed

Algorithm 1 Follow the Regularized Leader (FTRL)

- 1: Input: convex compact decision set $\mathcal{K} \subset \mathbb{R}^N$
- 2: Input: strictly convex differentiable regularization function $\mathcal{R}(\cdot)$ defined on \mathcal{K}
- 3: Parameter: $\eta > 0$
- 4: Initialize: $\mathbf{L}_1 = \langle 0, \dots, 0 \rangle$
- 5: **for** $t = 1, \dots, T$ **do**
- 6: The learner selects action $\mathbf{w}_t \in \mathcal{K}$ according to:

$$\mathbf{w}_t := \operatorname{argmin}_{\mathbf{w} \in \mathcal{K}} \mathbf{L}_{t-1} \cdot \mathbf{w} + \frac{1}{\eta} \mathcal{R}(\mathbf{w}) \quad (10)$$

- 7: Nature reveals ℓ_t , learner suffers loss $\ell_t \cdot \mathbf{w}_t$
 - 8: The learner updates $\mathbf{L}_t = \mathbf{L}_{t-1} + \ell_t$
 - 9: **end for**
-

bounded convex action space $\mathcal{K} \subset \mathbb{R}^N$. Notice that the experts setting is just a special case of online linear optimization, where the set \mathcal{K} is precisely the N -simplex Δ_N .

For a complete description of the FTRL algorithm, we refer the reader to the excellent notes of Alexander Rakhlin [54]. We will make use of a result from these notes, but we first include two additional assumptions that we will use to make the connection to duality-based cost function market makers. In the remainder of this section, we use $\|\cdot\|$ to denote the L2 norm.

Assumption 1. For each time step t , $\|\ell_t\| \leq 1$.

Assumption 2. The regularizer $\mathcal{R}(\cdot)$ has the Legendre property defined in Section 11.2 of Cesa-Bianchi and Lugosi [13]: \mathcal{R} is strictly convex on $\operatorname{relint}(\mathcal{K})$ and $\|\nabla \mathcal{R}(\mathbf{w})\| \rightarrow \infty$ as $\mathbf{w} \rightarrow \operatorname{relbnd}(\mathcal{K})$.

Under the latter assumption, the solution to Equation 10 will always occur in the relative interior of \mathcal{K} , which implies that the optimization is effectively unconstrained. We can now utilize Corollary 9 of Rakhlin [54] to obtain the following.

Proposition 1. Under Assumptions 1 and 2, the FTRL algorithm enjoys the following regret bound: For any $\mathbf{w}^* \in \mathcal{K}$,

$$\sum_{t=1}^T \ell_t \cdot \mathbf{w}_t - \sum_{t=1}^T \ell_t \cdot \mathbf{w}^* \leq \frac{1}{\eta} \left(\mathcal{R}(\mathbf{w}^*) - \mathcal{R}(\mathbf{w}_1) - D_{\mathcal{R}}(\mathbf{w}^*, \mathbf{w}_{T+1}) + \sum_{t=1}^T D_{\mathcal{R}}(\mathbf{w}_t, \mathbf{w}_{t+1}) \right).$$

This proposition may not be so illuminating at first glance, but it expresses a fundamental tradeoff in the learning problem. If we choose a regularizer \mathcal{R} with heavy curvature, or equivalently if we choose a small η , then given the nature of the optimization problem in Equation 10, we ensure that the updates $\mathbf{w}_t \rightarrow \mathbf{w}_{t+1}$ are “small” and hence $D_{\mathcal{R}}(\mathbf{w}_t, \mathbf{w}_{t+1})$ will be small. On the other hand, we pay for either of these choices since (a) the bound is proportional to $1/\eta$, and (b) the difference $\mathcal{R}(\mathbf{w}^*) - \mathcal{R}(\mathbf{w}_1)$ grows larger when \mathcal{R} has more curvature.

Under certain reasonable assumptions on \mathcal{R} , it is possible to prove that $D_{\mathcal{R}}(\mathbf{w}_t, \mathbf{w}_{t+1}) \leq O(\eta^2)$. For example, if \mathcal{R} is strongly convex (with respect to the L2 norm), then $D_{\mathcal{R}}(\mathbf{w}_t, \mathbf{w}_{t+1}) \leq \eta^2 \|\ell_t\|^2$. See Rakhlin [54] for more details.

Corollary 1. *Suppose that there exists $B > 0$ such that for every t , $D_{\mathcal{R}}(\mathbf{w}_t, \mathbf{w}_{t+1}) \leq B\eta^2$, and that there exists $C > 0$ such that $\mathcal{R}(\mathbf{w}^*) - \mathcal{R}(\mathbf{w}_1) \leq C$. Then $\text{Regret}(\text{FTRL}) \leq C/\eta + \eta BT$. If $\eta = \sqrt{C/BT}$, then $\text{Regret}(\text{FTRL}) \leq 2\sqrt{BCT}$.*

This final bound is quite powerful. It says that the regret of any online linear optimization problem will always be on the order of \sqrt{T} , and the constant in front of this rate will depend on the total variation of regularization function on \mathcal{K} (that is, $\mathcal{R}(\mathbf{w}^*) - \mathcal{R}(\mathbf{w}_1)$) as well as the stability of the updates (that is, the terms $D_{\mathcal{R}}(\mathbf{w}_t, \mathbf{w}_{t+1})$).

7.2 An Equivalence Between Online Learning and Market Making

Having reviewed a much of the literature on the design of online learning algorithms, we now pivot back to the the primary topic at hand, the design of market makers for complex security spaces. We will see that the tools that have been developed for the online learning setting are strikingly similar to those we have constructed for selecting pricing mechanisms. This is rather surprising, as the problem of learning in an online environment is semantically quite distinct from the problem of pricing securities in a prediction market: a learning algorithm receives *losses* and selects *weights* whereas a market maker manages *trades* and sets *prices*. We now show how these two problems can be viewed as two sides of the same coin. The two frameworks have very different semantics yet, in a very strong sense, have nearly identical *syntax*.

The relationship is described in full detail in Figure 1. We imagine that the learner uses the FTRL algorithm (Algorithm 1) to select weights, and the prediction market uses our market making framework.

What we emphasize in Figure 1 is that, by identifying the objects Π , $R(\cdot)$, and $\{\mathbf{r}_t\}$ with the objects \mathcal{K} , $\mathcal{R}(\cdot)/\eta$, and $\{-\ell_t\}$, respectively, the mechanisms for choosing an instantaneous price vector $\mathbf{x}_t \in \Pi$ and selecting a weight vector $\mathbf{w}_t \in \mathcal{K}$ are identical. Put another way, if we consider security bundles \mathbf{r}_t as the negative loss vectors ℓ_t , then the market mechanism we have proposed (Section 3) becomes exactly FTRL (Algorithm 1).

The connection seems to break down when we arrive at the last pair of statements, as the FTRL regret and the market maker's worst-case loss do not appear to be identical. Strictly speaking this is true, however these two quantities are not so far apart. Using the previous identification, we see that the term $\max_{\mathbf{x} \in \Pi} \mathbf{x} \cdot \mathbf{q}_T$, representing the worst-case payout of the market maker, matches exactly the term $-\min_{\mathbf{w} \in \mathcal{K}} \mathbf{w} \cdot L_T$. Now let us do a first-order approximation on the first term, which is the market maker's earnings from selling securities:

$$C(\mathbf{q}_T) - C(\mathbf{q}_0) = \sum_{t=1}^T C(\mathbf{q}_{t+1}) - C(\mathbf{q}_t) \approx \sum_{t=1}^T \nabla C(\mathbf{q}_t) \cdot (\mathbf{q}_{t+1} - \mathbf{q}_t) = \sum_{t=1}^T \mathbf{x}_t \cdot \mathbf{r}_t \quad (11)$$

where we use the fact that the instantaneous price vector \mathbf{x}_t is equal to $\nabla C(\mathbf{q}_t)$. The latter quantity is not surprising, as every trader will roughly pay the instantaneous prices \mathbf{x}_t for the securities times the quantities \mathbf{r}_t of each security sold. The total earned by the market maker $C(\mathbf{q}_T) - C(\mathbf{q}_0)$ is then roughly the sum of these payments over all trades.

How bad is this approximation? We can quantify this explicitly, since the difference between $C(\mathbf{q}_{t+1}) - C(\mathbf{q}_t)$ and $\nabla C(\mathbf{q}_t) \cdot (\mathbf{q}_{t+1} - \mathbf{q}_t)$ is exactly the value $D_C(\mathbf{q}_{t+1}, \mathbf{q}_t)$. If R has the Legendre property (described in Assumption 2) then via standard arguments we can also conclude that $D_C(\mathbf{q}_{t+1}, \mathbf{q}_t) = D_R(\mathbf{x}_t, \mathbf{x}_{t+1})$. Under this assumption, in other words, the worst case loss of the market maker can be written as

$$\max_{\mathbf{x} \in \Pi} \mathbf{x} \cdot \mathbf{q}_T - \sum_{t=1}^T \mathbf{x}_t \cdot \mathbf{r}_t - \sum_{t=1}^T D_R(\mathbf{x}_t, \mathbf{x}_{t+1}).$$

In a given online linear optimization problem:

- the learner is given access to a fixed space of weights \mathcal{K}
- the learning algorithm must select a weight vector $\mathbf{w} \in \mathcal{K}$
- the learner uses a convex regularizer $\mathcal{R}(\cdot)$, which is a parameter of FTRL
- the learner receives loss vectors ℓ_t
- the learning algorithm maintains a cumulative loss vector \mathbf{L}_t and updates according to

$$\mathbf{L}_{t+1} \leftarrow \mathbf{L}_t + \ell_t$$

- FTRL selects the weight vector by solving

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

- the learner suffers regret

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

In a given prediction market with a complex security space:

- the market maker has an outcome space \mathcal{O} and a payoff function $\rho : \mathcal{O} \rightarrow \mathbb{R}^k$, which define a feasible price space $\Pi = \mathcal{H}(\rho(\mathcal{O}))$
- the market maker must select instantaneous security prices $\mathbf{x} \in \Pi$
- the market maker uses a convex conjugate $R(\cdot)$, which is a parameter of the pricing function $C(\cdot)$
- the market maker receives security bundle purchases \mathbf{r}_t
- the market maker maintains a quantity vector \mathbf{q}_t and updates according to

$$\mathbf{q}_{t+1} \leftarrow \mathbf{q}_t + \mathbf{r}_t$$

- the market mechanism sets prices via

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

- the market maker suffers worst-case loss

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

Figure 1: The similarities between our market making framework and the Follow the Regularized Leader algorithm for online linear optimization.

Putting everything together, this final bound is exactly what we should expect. Look again at Theorem 9 and Proposition 1. The bounds in these theorems are nearly identical under the translation matching $\mathbf{w}^* \leftrightarrow \boldsymbol{\rho}(\mathbf{o})$, $\mathbf{w}_{T+1} \leftrightarrow \nabla C(\mathbf{q})$, and $R(\mathbf{x}) \leftrightarrow \mathcal{R}(\mathbf{w})/\eta$, since by definition of FTRL, $\mathbf{w}_1 = \arg \min_{\mathbf{w} \in \mathcal{K}} \mathcal{R}(\mathbf{w})$. The key difference is that the sum of divergence terms $\sum_{t=1}^T D_{\mathcal{R}}(\mathbf{w}_t, \mathbf{w}_{t+1})$ seems to get “lost in translation” when we look at Theorem 9. The above equation tells us why this is.

It is worth looking further into this key difference between the FTRL algorithm for online linear optimization and our proposed automated market maker. We could imagine a modified market maker with a different mechanism: after the $(t - 1)$ th trade the market maker posts the (instantaneous) price vector \mathbf{x}_t , a trader arrives to purchase bundle \mathbf{r}_t , and the trader pays exactly $\mathbf{x}_t \cdot \mathbf{r}_t$. Notice this is different from the original framework, where the trader would pay $C(\mathbf{q} + \mathbf{r}_t) - C(\mathbf{q})$, although we observed in (11) that these two values are not so far apart.

Notice that under the mapping outlined in Figure 1, algorithms for the expert setting ($\mathcal{K} = \Delta_n$) correspond to complete markets. Weighted Majority corresponds directly to LMSR, with the learning rate η playing a similar rule to the LMSR parameter b . The similarity between the Weighted Majority weights (Equation 9) and the LMSR prices (Equation 2) has been observed and exploited in the past [18]. The Quad-SCPM market [4] can be mapped to online gradient descent, which is known to be equivalent to FTRL with a quadratic regularizer [34, 36].

8 Relation to Market Scoring Rules

We have described ways in which our optimization-based framework can be used to derive novel, efficient automated market makers for markets in which the outcome space is very large. Our framework also provides new insights into the complete market setting. In this section, we describe how our framework can be used to establish a correspondence between cost function based markets and market scoring rules.

Consider the special case of a complete market. For complete markets that offer n Arrow-Debreu securities for the n mutually exclusive and exhaustive outcomes, our framework defines a set of market makers by equating the set of allowable prices Π to the n -simplex. That is, a market maker for a complete market that satisfies conditions 2-5 in Section 3 can use a cost function

$$C(\mathbf{q}) = \sup_{\mathbf{x} \in \Delta_n} \mathbf{x} \cdot \mathbf{q} - R(\mathbf{x}), \quad (12)$$

where $R(\mathbf{x})$ is strictly convex over Δ_n . The market price $\mathbf{x}(\mathbf{q}) = \nabla C(\mathbf{q})$ is the optimal solution to the convex optimization. It is easy to check that when $R(x) = b \sum_{i=1}^n x_i \log x_i$, the negative entropy function, we have the LMSR market maker. The LMSR is a popular example of a large class of market makers, called *market scoring rules* (MSRs). In this section, after reviewing the notion of a proper scoring rule and describing the class of MSRs, we use Equation 12 to establish a correspondence between MSRs and cost function based market makers for complete markets.

8.1 Proper Scoring Rules

Scoring rules have long been used in the evaluation of probabilistic forecasts. In the context of information elicitation, scoring rules are used to encourage individuals to make careful assessments and truthfully report their beliefs [25, 41, 57]. In the context of machine learning, scoring rules are used as loss functions to evaluate and compare the performance of different algorithms [12, 55]. We briefly mention recent work of Abernethy and Frongillo [1] who used a generalized notion of a scoring rule in order to construct a market mechanism for solving machine learning problems.

Formally, let $\{1, \dots, n\}$ be a set of mutually exclusive and exhaustive outcomes of a future event. A scoring rule s maps a probability distribution \mathbf{p} over outcomes to a score $s_i(\mathbf{p})$ for each outcome i , with $s_i(\mathbf{p})$ taking values in the range $[-\infty, \infty]$. Intuitively, this score represents the reward that a forecaster receives for predicting the distribution \mathbf{p} if the outcome turns out to be i . A scoring rule is said to be *regular* relative to the probability simplex Δ_n if $\sum_{i=1}^n p_i s_i(\mathbf{p}') \in [-\infty, \infty)$ for all $\mathbf{p}, \mathbf{p}' \in \Delta_n$, with $\sum_{i=1}^n p_i s_i(\mathbf{p}) \in (-\infty, \infty)$. This implies that $s_i(\mathbf{p})$ is finite whenever $p_i > 0$. A scoring rule is said to be *proper* if a risk-neutral forecaster who believes the true distribution over outcomes to be \mathbf{p} has no incentive to report any alternate distribution \mathbf{p}' , that is, if $\sum_{i=1}^n p_i s_i(\mathbf{p}) \geq \sum_{i=1}^n p_i s_i(\mathbf{p}')$ for all distributions \mathbf{p}' . The rule is *strictly proper* if this inequality holds with equality only when $\mathbf{p} = \mathbf{p}'$.

Two examples of regular, strictly proper scoring rules commonly used in both information elicitation and machine learning are the the quadratic scoring rule [11]:

$$s_i(\mathbf{p}) = a_i + b \left(2p_i - \sum_{i=1}^n p_i^2 \right) \quad (13)$$

and the logarithmic scoring rule [28]:

$$s_i(\mathbf{p}) = a_i + b \log(p_i) \quad (14)$$

where $b > 0$ and a_1, \dots, a_n are parameters.

Proper scoring rules are closely related to convex functions. In fact, the following characterization theorem of Gneiting and Raftery [27], which is credited to McCarthy [46] and Savage [57], gives the precise relationship between convex functions and proper scoring rules.

Theorem 11 (Gneiting and Raftery [27]). *A regular scoring rule is (strictly) proper if and only if there exists a (strictly) convex function $G : \Delta_n \rightarrow \mathbb{R}$ such that for all $i \in \{1, \dots, n\}$,*

$$s_i(\mathbf{p}) = G(\mathbf{p}) - G'(\mathbf{p}) \cdot \mathbf{p} + G'_i(\mathbf{p}),$$

where $G'(\mathbf{p})$ is any subgradient of G at the point \mathbf{p} and $G'_i(\mathbf{p})$ is the i -th element of $G'(\mathbf{p})$.

Note that for a scoring rule defined in terms of a function G ,

$$\sum_{i=1}^n p_i s_i(\mathbf{p}) = \sum_{i=1}^n p_i (G(\mathbf{p}) - G'(\mathbf{p}) \cdot \mathbf{p} + G'_i(\mathbf{p})) = G(\mathbf{p}).$$

Theorem 11 therefore indicates that a regular scoring rule is (strictly) proper if and only if its expected score function $G(\mathbf{p})$ is (strictly) convex on Δ_n , and the vector with elements $s_i(\mathbf{p})$ is a subgradient of G at the point \mathbf{p} . Hence, every bounded convex function G over Δ_n induces a proper scoring rule.

Define $S(\tilde{\mathbf{p}}, \mathbf{p}) = \sum_{i=1}^n p_i s_i(\tilde{\mathbf{p}})$ to be the expected score of a forecaster who has belief \mathbf{p} but predicts $\tilde{\mathbf{p}}$. Then, $G(\mathbf{p}) = S(\mathbf{p}, \mathbf{p})$. If a scoring rule is regular and proper, $d(\tilde{\mathbf{p}}, \mathbf{p}) = S(\mathbf{p}, \mathbf{p}) - S(\tilde{\mathbf{p}}, \mathbf{p})$ is the associated *divergence function* that captures the expected score loss if a forecaster predicts $\tilde{\mathbf{p}}$ rather than his true belief \mathbf{p} . It is known that if $G(\mathbf{p})$ is differentiable, the divergence function is the Bregman divergence for G , that is, $d(\tilde{\mathbf{p}}, \mathbf{p}) = D_G(\tilde{\mathbf{p}}, \mathbf{p})$. For a nice survey on uses, properties, and characterizations of proper scoring rules, see Gneiting and Raftery [27].

8.2 Market Scoring Rules

Market scoring rules (MSR) were developed by Hanson [32, 33] as a method of using scoring rules to pool opinions from many different forecasters. Market scoring rules are sequentially shared scoring rules. Formally, the market maintains a current probability distribution \mathbf{p} . At any time, a trader can enter the market and change this distribution to an arbitrary distribution \mathbf{p}' of her choice.¹² If the outcome turns out to be i , she receives a (possibly negative) payoff of $s_i(\mathbf{p}') - s_i(\mathbf{p})$. For example, in the MSR defined using the logarithmic scoring rule in Equation 14, a trader who changes the distribution from \mathbf{p} to \mathbf{p}' receives a payoff of $b \log(p'_i/p_i)$. This market formulation is equivalent to the cost function based formulation of the LMSR (hence its name) in the sense that a trader who changes the market probabilities from \mathbf{p} to \mathbf{p}' in the MSR formulation receives the same payoff for every outcome i as a trader who changes the quantity vectors from any \mathbf{q} to \mathbf{q}' such that market prices become $\mathbf{x}(\mathbf{q}) = \mathbf{p}$ and $\mathbf{x}(\mathbf{q}') = \mathbf{p}'$ in the cost function based formulation. Using proper scoring rules, market scoring rules preserve the nice incentive compatible property of proper scoring rules for *myopic* traders. A trader who believes the true distribution to be \mathbf{p} and only cares about payoff of her current action, maximizes her expected payoff by changing the market's distribution to \mathbf{p} .

One advantage of the market scoring formulation is the ease of bounding the market maker's worst case loss. Each trader in a market scoring rule is essentially responsible for paying the previous trader's score. Thus the market maker is responsible only for paying the score of the final trader. Let \mathbf{p}_0 be the initial probability distribution of the market. The worst case loss of the market maker is then

$$\max_{i \in \{1, \dots, N\}} \sup_{\mathbf{p} \in \Delta_n} (s_i(\mathbf{p}) - s_i(\mathbf{p}_0)).$$

The Logarithmic Market Scoring Rule market maker is not the only market that can be defined as either a market scoring rule or a cost function based market. The fact that there exists a correspondence between certain market scoring rules and certain cost function based markets was noted by Chen and Pennock [14]. They pointed out that the MSR with scoring function \mathbf{s} and the cost function based market with cost function C are equivalent if for all \mathbf{q} and all outcomes i , $C(\mathbf{q}) = q_i - s_i(\mathbf{x}(\mathbf{q}))$. However, they provide neither any guarantees about the circumstances under which this condition can be satisfied nor a general way to find the cost function given a market scoring rule because $\mathbf{x}(\mathbf{q})$ is the gradient of $C(\mathbf{q})$ and the condition defines a differential equation. Agrawal et al. [4] also made use of the equivalence between markets when this strong condition holds. In the next section, we will give very general precise conditions under which an MSR is equivalent to a cost function based market and provide a way to translate a market scoring rule to a cost function based market and vice versa.

8.3 Equivalence between Market Scoring Rules and Cost Function Based Market Makers

Recall that a convex cost function C can be defined as $C(\mathbf{q}) = \sup_{\mathbf{x} \in \Delta_n} \sum_{i=1}^N x_i q_i - R(\mathbf{x})$ for a strictly convex function R . Let R_C denote the function R corresponding to the cost function C . According to Theorem 11, there is a one-to-one and onto mapping between strictly convex and differentiable R_C and strictly proper, regular scoring rules with differentiable scoring functions $s_i(\mathbf{x})$, where for every pair we have

$$R_C(\mathbf{x}) = \sum_{i=1}^n x_i s_i(\mathbf{x}), \quad (15)$$

¹²In some market scoring rules, such as the LMSR, distributions that place a weight of 0 on any outcome are not allowed since a trader would have to pay an infinite amount of money if the outcome with reported probability 0 actually occurred.

and

$$s_i(\mathbf{x}) = R_C(\mathbf{x}) - \sum_{j=1}^n \frac{\partial R_C(\mathbf{x})}{\partial x_j} x_j + \frac{\partial R_C(\mathbf{x})}{\partial x_i}. \quad (16)$$

Theorem 12 below shows that that the cost function based market using R_C in (15) and the market scoring rule market using $s_i(x)$ in (16) are equivalent in terms of trader's profit and reachable price vectors.

Theorem 12. *Given a pair of strictly convex, differentiable $R_C(x)$ and strictly proper, regular scoring rule with differentiable scoring functions $s_i(\mathbf{x})$ as defined in (15) and (16) respectively, the corresponding cost function based market and market scoring rule market are equivalent in the following two aspects:*

- *The profit of a trade is the same in the two markets if the trade starts with the same market prices and results in the same market prices and the prices for all outcomes are positive before and after the trade.*
- *Every price vector \mathbf{p} achievable in the market scoring rule is achievable in the cost function based market.*

Proof. We first show that a trader gets exactly the same profit for any realized outcome in the two markets if the market probabilities are positive. In a cost function based market, we know that $C(\mathbf{q}) = \sup_{x \in \Delta_n} \sum_{i=1}^n x_i q_i - R_C(\mathbf{x})$ and the market price $\mathbf{x}(\mathbf{q}) = \operatorname{argmax}_{\mathbf{x} \in \Delta_n} (\sum_{i=1}^n x_i q_i - R_C(\mathbf{x}))$. If $\mathbf{x}(\mathbf{q}) > \mathbf{0}$, the optimal solution must also be optimal for the unconstrained optimization problem $\max_{\mathbf{x}} \sum_{i=1}^n x_i q_i - R_C(\mathbf{x}) - \lambda_{\mathbf{q}} (\sum_{i=1}^n x_i - 1)$ for some $\lambda_{\mathbf{q}}$. Hence, we have $q_i = \frac{\partial R_C(\mathbf{x}(\mathbf{q}))}{\partial x_i(\mathbf{q})} + \lambda_{\mathbf{q}}$. Suppose in the cost function based market a trader changes the outstanding shares from \mathbf{q} to \mathbf{q}' and this trade changes the market price from $\mathbf{x}(\mathbf{q}) > \mathbf{0}$ to $\mathbf{x}(\mathbf{q}') > \mathbf{0}$. If outcome i occurs, the trader's profit is

$$\begin{aligned} & (q'_i - q_i) - (C(\mathbf{q}') - C(\mathbf{q})) \\ &= (q'_i - q_i) - \left(\sum_{j=1}^n x_j(\mathbf{q}') q'_j - R_C(\mathbf{x}(\mathbf{q}')) \right) + \left(\sum_{j=1}^n x_j(\mathbf{q}) q_j - R_C(\mathbf{x}(\mathbf{q})) \right) \\ &= \left(q'_i - \sum_{j=1}^n x_j(\mathbf{q}') q'_j + R_C(\mathbf{x}(\mathbf{q}')) \right) - \left(q_i - \sum_{j=1}^n x_j(\mathbf{q}) q_j + R_C(\mathbf{x}(\mathbf{q})) \right) \\ &= \left(\frac{\partial R_C(\mathbf{x}(\mathbf{q}'))}{\partial x_i(\mathbf{q}')} - \sum_{j=1}^n x_j(\mathbf{q}') \frac{\partial R_C(\mathbf{x}(\mathbf{q}'))}{\partial x_j(\mathbf{q}')} + R_C(\mathbf{x}(\mathbf{q}')) \right) \\ &\quad - \left(\frac{\partial R_C(\mathbf{x}(\mathbf{q}))}{\partial x_i(\mathbf{q})} - \sum_{j=1}^n x_j(\mathbf{q}) \frac{\partial R_C(\mathbf{x}(\mathbf{q}))}{\partial x_j(\mathbf{q})} + R_C(\mathbf{x}(\mathbf{q})) \right) \\ &= s_i(\mathbf{x}(\mathbf{q}')) - s_i(\mathbf{x}(\mathbf{q})). \end{aligned}$$

The first equality is due to $\mathbf{x}(\mathbf{q}) = \operatorname{argmax}_{\mathbf{x} \in \Delta_n} (\sum_{i=1}^n x_i q_i - R_C(\mathbf{x}))$, and the third equality is due to $q_i = \frac{\partial R_C(\mathbf{x}(\mathbf{q}))}{\partial x_i(\mathbf{q})} + \lambda_{\mathbf{q}}$.

Next, observe that using the cost function based market it is possible to achieve any price vector \mathbf{x} with finite scores $s_i(\mathbf{x})$ by setting $q_i = s_i(\mathbf{x})$ for all i , because $\mathbf{x} = \operatorname{argmax}_{\mathbf{x}' \in \Delta_n} \sum_{i=1}^n x'_i s_i(\mathbf{x}) - \sum_{i=1}^n x'_i s_i(\mathbf{x}')$. Since \mathbf{s} is regular, this implies that it is possible to achieve any prices in the interior of the probability simplex using the cost function based market (and any prices \mathbf{x} on the exterior as long as $s_i(\mathbf{x})$ is finite for all i). \square

9 Conclusion

We conclude by mentioning one nice direction of work. As we discussed, there is an inherent tradeoff between the bid-ask spread and the worst-case loss of the market maker. But if the market maker chooses to sell securities with an additional *transaction cost* for each security sold, then this money can not only help to cover the worst-case loss, but can also lead to a profit. Furthermore, if a market becomes popular, the market-maker may wish to increase the market depth. This idea has been explored by Othman et al. [49] for the case of complete markets, introducing a *liquidity sensitive* market maker, and they provide a new model with profit guarantees. Othman and Sandholm [48] recently extend the work and characterize a family of market makers that are liquidity sensitive. Via our framework, we can define an alternative method for simultaneously including transaction costs and guaranteeing profit. In particular, this is achieved through relaxing the price space, as discussed in Section 6.2. We leave the details to future work.

10 Acknowledgments

This material is based upon work supported by NSF under CNS-0937060 to the CRA for the CIFellows Project, NSF grants CCF-0953516, CCF-0915016, IIS-1054911, and DMS-070706, DARPA grant FA8750-05-2-0249, and a Yahoo! PhD Fellowship. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors alone. The authors are grateful to David Pennock for useful discussions about this work.

References

- [1] J. Abernethy and R. M. Frongillo. A collaborative mechanism for crowdsourcing prediction problems. In *Advances in Neural Information Processing Systems*, 2011.
- [2] J. Abernethy, Y. Chen, and J. Wortman Vaughan. An optimization-based framework for automated market-making. In *Proceedings of the 12th ACM Conference on Electronic Commerce*, 2011.
- [3] S. Agrawal, Z. Wang, and Y. Ye. Parimutuel betting on permutations. In *Proceedings of the 4th International Workshop On Internet And Network Economics*, 2008.
- [4] S. Agrawal, E. Delage, M. Peters, Z. Wang, and Y. Ye. A unified framework for dynamic prediction market design. *Operations Research*, 59(3):550–568, 2011.
- [5] K. J. Arrow. The role of securities in the optimal allocation of risk-bearing. *Review of Economic Studies*, 31(2): 91–96, 1964.
- [6] K. J. Arrow. *Essays in the Theory of Risk Bearing*. North Holland, Amsterdam, 1970.
- [7] K.S. Azoury and M.K. Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 43(3):211–246, 2001.
- [8] 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*. 2001.
- [9] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [10] A. Brahma, S. Das, and M. Magdon-Ismail. Comparing prediction market structures, with an application to market making. Working paper, September 2010.

- [11] G. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950.
- [12] A. Buja, W. Stuetzle, and Y. Shen. Loss functions for binary class probability estimation and classification: Structure and applications. Working draft, November 2005.
- [13] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.
- [14] Y. Chen and D. M. Pennock. A utility framework for bounded-loss market makers. In *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence*, 2007.
- [15] Y. Chen and J. Wortman Vaughan. A new understanding of prediction markets via no-regret learning. In *Proceedings of the 11th ACM Conference on Electronic Commerce*, 2010.
- [16] Y. Chen, L. Fortnow, E. Nikolova, and D. M. Pennock. Betting on permutations. In *Proceedings of the 8th ACM conference on Electronic commerce*, pages 326–335, 2007.
- [17] Y. Chen, L. Fortnow, E. Nikolova, and D.M. Pennock. Betting on permutations. In *Proceedings of the 8th ACM Conference on Electronic Commerce*, pages 326–335. ACM, 2007.
- [18] Y. Chen, L. Fortnow, N. Lambert, D. M. Pennock, and J. Wortman. Complexity of combinatorial market makers. In *Proceedings of the 9th ACM Conference on Electronic Commerce*, 2008.
- [19] Y. Chen, S. Goel, and D. M. Pennock. Pricing combinatorial markets for tournaments. In *Proceedings of the 40th ACM Symposium on Theory of Computing*, 2008.
- [20] T.M. Cover. Universal portfolios. *Mathematical Finance*, 1(1):1–29, January 1991. URL citeseer.ist.psu.edu/article/cover96universal.html.
- [21] S. Das and M. Magdon-Ismail. Adapting to a market shock: Optimal sequential market-making. In *Proceedings of the 21th Annual Conference on Neural Information Processing Systems*, pages 361–368, 2008.
- [22] 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.
- [23] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Comp. and System Sciences*, 55(1):119–139, 1997.
- [24] X. Gao, Y. Chen, and D. M. Pennock. Betting on the real line. In *Proceedings of the 5th Workshop on Internet and Network Economics*, 2009.
- [25] P. H. Garthwaite, J. B. Kadane, and A. O’Hagan. Statistical methods for eliciting probability distributions. *Journal of the American Statistical Association*, 100:680–701, 2005.
- [26] M. Ghodsi, H. Mahini, V. S. Mirrokni, and M. Zadimoghaddam. Permutation betting markets: Singleton betting with extra information. In *Proceedings of the 9th ACM conference on Electronic commerce*, pages 180–189, 2008.
- [27] T. Gneiting and A. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.
- [28] I. J. Good. Rational decisions. *Journal of the Royal Statistical Society, Series B (Methodological)*, 14(1):107–114, 1952.
- [29] G. Gorni. Conjugation and second-order properties of convex functions. *Journal of Mathematical Analysis and Applications*, 158(2):293–315, 1991.

- [30] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981. ISSN 0209-9683.
- [31] M. Guo and D. M. Pennock. Combinatorial prediction markets for event hierarchies. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, 2009.
- [32] R. Hanson. Combinatorial information market design. *Information Systems Frontiers*, 5(1):105–119, 2003.
- [33] R. Hanson. Logarithmic market scoring rules for modular combinatorial information aggregation. *Journal of Prediction Markets*, 1(1):3–15, 2007.
- [34] E. Hazan. A survey: The convex optimization approach to regret minimization. Draft, Sept. 2009.
- [35] E. Hazan and S. Kale. Extracting certainty from uncertainty: Regret bounded by variation in costs. In *COLT*, 2008.
- [36] E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2–3):169–192, 2007.
- [37] D. Helmbold and M. Warmuth. Learning permutations with exponential weights. *JMLR*, 10:1705–1736, 2009.
- [38] J.-B. Hiriart-Urruty and C. Lemaréchal. *Fundamentals of Convex Analysis*. Springer, 2001.
- [39] J. Kivinen and M. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Journal of Information and Computation*, 132(1):1–63, 1997.
- [40] J. Kivinen and M. Warmuth. Averaging expert predictions. In *EuroCOLT*, 1999.
- [41] N. Lambert, D. M. Pennock, and Y. Shoham. Eliciting properties of probability distributions. In *Proceedings of the 9th ACM Conference on Electronic Commerce*, 2008.
- [42] J. Ledyard, R. Hanson, and T. Ishikida. An experimental test of combinatorial information markets. *Journal of Economic Behavior and Organization*, 69:182–189, 2009.
- [43] N. Littlestone and M. Warmuth. The weighted majority algorithm. *Info. and Computation*, 108(2):212–261, 1994.
- [44] B. Mangold, M. Dooley, R. Dornfest, G. W. Flake, H. Hoffman, T. Kasturi, and D. M. Pennock. The tech buzz game. *IEEE Computer*, 38(7):94–97, 2005.
- [45] A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomics Theory*. Oxford University Press, New York, NY, 1995.
- [46] J. McCarthy. Measures of the value of information. *PNAS*, 42:654–655, 1956.
- [47] N. Megiddo. Mixtures of order matrices and generalized order matrices. *Discrete Mathematics*, 19(2):177–181, 1977. ISSN 0012-365X.
- [48] A. Othman and T. Sandholm. Homogeneous risk measures and liquidity-sensitive automated market makers. Carnegie Mellon University working paper, March 2011.
- [49] A. Othman, T. Sandholm, D. M. Pennock, and D. M. Reeves. A practical liquidity-sensitive automated market maker. In *Proceedings of the 11th ACM Conference on Electronic Commerce*, 2010.
- [50] D. M. Pennock. A dynamic pari-mutuel market for hedging, wagering, and information aggregation. In *Proceedings of the Fifth ACM Conference on Electronic Commerce (EC'04)*, 2004.

- [51] D. M. Pennock and R. Sami. Computational aspects of prediction markets. In N. Nisan, T. Roughgarden, É. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*. Cambridge University Press, 2007.
- [52] D. M. Pennock and L. Xia. Price updating in combinatorial prediction markets with bayesian networks. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence*, pages 581–588, 2011.
- [53] M. Peters, A. Man-Cho So, and Y. Ye. Pari-mutuel markets: Mechanisms and performance. In *Proceedings of the 3rd International Workshop on Internet and Network Economics*, pages 82–95, 2007.
- [54] A. Rakhlin. Lecture notes on online learning. Draft, April 2009.
- [55] M. D. Reid and R. C. Williamson. Surrogate regret bounds for proper losses. In *ICML*, 2009.
- [56] R.T. Rockafellar. *Convex analysis*. Princeton Univ Press, 1970.
- [57] L. J. Savage. Elicitation of personal probabilities and expectations. *Journal of the American Statistical Association*, 66(336):783–801, 1971.
- [58] S. Shalev-Shwartz and Y. Singer. A primal-dual perspective of online learning algorithms. *Machine Learning*, 69(2–3):115–142, 2007.
- [59] M. Sion. On general minimax theorems. *Pacific Journal of Mathematics*, 8(1):171–176, 1958.
- [60] J. Wolfers and E. Zitzewitz. Prediction markets. *Journal of Economic Perspective*, 18(2):107–126, 2004.
- [61] L. Xia and D. M. Pennock. An efficient monte-carlo algorithm for pricing combinatorial prediction markets for tournaments. In *Proceedings of the International Joint Conferences on Artificial Intelligence*, 2011.
- [62] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, 2003.