



A New Understanding of Prediction Markets Via No-Regret Learning

Citation

Yiling, Chen and Jennifer Wortman Vaughan. 2010. A new understanding of prediction markets via no-regret learning. In 11th ACM Conference on Electronic Commerce: June 7-11, 2010, Cambridge, MA, 189-198. New York, NY: Association for Computing Machinery.

Published Version

doi:10.1145/1807342.1807372

Permanent link

<http://nrs.harvard.edu/urn-3:HUL.InstRepos:8896228>

Terms of Use

This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Open Access Policy Articles, as set forth at <http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#OAP>

Share Your Story

The Harvard community has made this article openly available.
Please share how this access benefits you. [Submit a story](#).

[Accessibility](#)

A New Understanding of Prediction Markets Via No-Regret Learning

Yiling Chen

School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138
yiling@eecs.harvard.edu

Jennifer Wortman Vaughan

School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138
jenn@seas.harvard.edu

ABSTRACT

We explore the striking mathematical connections that exist between market scoring rules, cost function based prediction markets, and no-regret learning. We first show that any cost function based prediction market can be interpreted as an algorithm for the commonly studied problem of learning from expert advice by equating the set of *outcomes* on which bets are placed in the market with the set of *experts* in the learning setting, and equating *trades* made in the market with *losses* observed by the learning algorithm. If the loss of the market organizer is bounded, this bound can be used to derive an $O(\sqrt{T})$ regret bound for the corresponding learning algorithm. We then show that the class of markets with convex cost functions exactly corresponds to the class of Follow the Regularized Leader learning algorithms, with the choice of a cost function in the market corresponding to the choice of a regularizer in the learning problem. Finally, we show an equivalence between market scoring rules and prediction markets with convex cost functions. This implies both that any market scoring rule can be implemented as a cost function based market maker, and that market scoring rules can be interpreted naturally as Follow the Regularized Leader algorithms. These connections provide new insight into how it is that commonly studied markets, such as the Logarithmic Market Scoring Rule, can aggregate opinions into accurate estimates of the likelihood of future events.

Categories and Subject Descriptors

J.4 [Social and Behavioral Sciences]: Economics; I.2.6 [Artificial Intelligence]: Learning

General Terms

Algorithms, Economics, Theory

1. INTRODUCTION

Imagine you are interested in learning an accurate estimate of the probability that the United States unemployment rate for a particular month will fall below 10%. You

could choose to spend hours digging through news articles, reading financial reports, and weighing various opinions against each other, eventually coming up with a reasonably informed estimate. However, you could potentially save yourself a lot of hassle (and obtain a better estimate!) by appealing to the wisdom of crowds.

A *prediction market* is a financial market designed for information aggregation. For example, in a cost function based prediction market [5], the organizer (or *market maker*) trades a set of securities corresponding to each potential outcome of an event. The market maker might offer a security that pays \$1 if and only if the U.S. unemployment rate for January 2010 is above 10%. A risk neutral trader who believes that the true probability that the unemployment rate will be above 10% is p should be willing to buy a share of this security at any price below $\$p$. Similarly, he should be willing to sell a share of this security at any price above $\$p$. For this reason, the current market price of this security can be viewed as the population's collective estimate of how likely it is that the unemployment rate will be above 10%.

These estimates have proved quite accurate in practice in a wide variety of domains. (See Ledyard et al. [27] for an impressive assortment of examples.) The theory of rational expectations equilibria offers some insight into why prediction markets in general should converge to accurate prices, but is plagued by strong assumptions and no-trade theorems [31]. Furthermore, this theory says nothing of why particular prediction market mechanisms, such as Hanson's increasingly popular Logarithmic Market Scoring Rule (LMSR) [15, 16], might produce more accurate estimates than others in practice. In this work, we aim to provide additional insight into the learning power of particular market mechanisms by highlighting the deep mathematical connections between prediction markets and no-regret learning.

It seems natural that there should be a connection between prediction markets and learning. The theories of markets and learning are built upon many of the same fundamental concepts, such as proper scoring rules (called proper losses in the learning community) and Bregman divergences. To our knowledge, Chen et al. [6] were the first to formally demonstrate a connection, showing that the standard Randomized Weighted Majority regret bound [9] can be used as a starting point to rederive the well-known bound on the worst-case loss of a LMSR market maker. (They went on to show that PermELearn, an extension of Weighted Majority to permutation learning [20], can be used to efficiently run LMSR over combinatorial outcome spaces for betting on rankings.) As we show in Section 4, the converse is also true;

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

EC'10, June 7–11, 2010, Cambridge, Massachusetts, USA.

Copyright 2010 ACM 978-1-60558-822-3/10/06 ...\$10.00.

the Weighted Majority regret bound can be derived directly from the bound on the worst-case loss of a market maker using LMSR. However, the connection goes much deeper.

In Section 4, we show how *any* cost function based prediction market with bounded loss can be interpreted as a no-regret learning algorithm. Furthermore, if the loss of the market maker is bounded, this bound can be used to derive an $O(\sqrt{T})$ regret bound for the corresponding learning algorithm. The key idea is to equate *outcomes* in the market setting with *experts* in the learning setting, and *trades* made in the market with *losses* observed by the learning algorithm.¹ We can then think of the market maker as learning a probability distribution over outcomes by treating each observed trade as a training instance, just as the learning algorithm learns a distribution over experts by observing losses.

In Section 5, we go on to show that the class of *convex* cost function based markets exactly corresponds to the class of Follow the Regularized Leader learning algorithms [34, 18, 17] in which weights are chosen at each time step to minimize a combination of empirical loss and a convex regularization term. This allows us to interpret the selection of a cost function for the market as the selection of a regularizer for the learning problem. Furthermore, we prove an equivalence between another common class of prediction markets, *market scoring rules*, and convex cost function based markets,² which immediately implies that market scoring rules can be interpreted as Follow the Regularized Leader algorithms too. These connections provide insight into why it is that prediction markets tend to yield such accurate estimates in practice.

We begin by reviewing the relevant concepts and results pertaining to prediction markets and no-regret learning.

2. PREDICTION MARKETS

A variety of compelling prediction market mechanisms have been proposed and studied, including standard call market mechanisms and dynamic parimutuel markets [30]. In this work we focus on two very general classes of mechanisms: Hanson’s market scoring rules [15, 16] and cost function based prediction markets as described in Chen and Pennock [5]. We also briefly discuss the related class of Sequential Convex Parimutuel Mechanisms [1] in Section 5.4.

2.1 Market Scoring Rules

Scoring rules have long been used in the evaluation of probabilistic forecasts. In the context of prediction markets and elicitation, scoring rules are used to encourage individuals to make careful assessments and truthfully report their beliefs [33, 11, 26]. In the context of machine learning, scoring rules are used as loss functions to evaluate and compare the performance of different algorithms [3, 32].

Formally, let $\{1, \dots, N\}$ be a set of mutually exclusive and exhaustive outcomes of a future event. A scoring rule \vec{s} maps a probability distribution \vec{p} over the outcomes to a score $s_i(\vec{p})$ for each outcome i , with $s_i(\vec{p})$ taking values in the extended real line $[-\infty, \infty]$. Intuitively, this score represents the reward of a forecaster might receive for predicting the distribution \vec{p} if the outcome turns out to be i . A

¹Note that it is *not* the traders in the market that are analogous to the experts, as one might initially guess.

²A similar but weaker correspondence between market scoring rules and cost function based markets was discussed in Chen and Pennock [5] and Agrawal et al. [1].

scoring rule is said to be *regular* relative to the probability simplex Δ_N if $\sum_{i=1}^N p_i s_i(\vec{p}') \in [-\infty, \infty)$ for all $\vec{p}, \vec{p}' \in \Delta_N$, with $\sum_{i=1}^N p_i s_i(\vec{p}) \in (-\infty, \infty)$. This implies that $s_i(\vec{p})$ 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 \vec{p} has no incentive to report any alternate distribution \vec{p}' , that is, if $\sum_{i=1}^N p_i s_i(\vec{p}) \geq \sum_{i=1}^N p_i s_i(\vec{p}')$ for all distributions \vec{p}' . The rule is *strictly proper* if this inequality holds with equality only when $\vec{p} = \vec{p}'$.

Two examples of regular, strictly proper scoring rules commonly used in both elicitation and in machine learning are the quadratic scoring rule [2]:

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

and the logarithmic scoring rule [13]:

$$s_i(\vec{p}) = a_i + b \log(p_i) \quad (2)$$

with parameters $b > 0$ and a_1, \dots, a_N . The uses and properties of scoring rules are too extensive to cover in detail here. For a nice survey, see Gneiting and Raftery [12].

Market scoring rules were developed by Hanson [15, 16] 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 \vec{p} . At any time, a trader can enter the market and change this distribution to an arbitrary distribution \vec{p}' of her choice.³ If the outcome turns out to be i , she receives a (possibly negative) payoff of $s_i(\vec{p}') - s_i(\vec{p})$. For example, in the popular Logarithmic Market Scoring Rule (LMSR), which is based on the logarithmic scoring rule in Equation 2, a trader who changes the distribution from \vec{p} to \vec{p}' receives a payoff of $b \log(p'_i/p_i)$.

Since the trader has no control over \vec{p} , a myopic trader who believes the true distribution to be \vec{r} maximizes her expected payoff by maximizing $\sum_i r_i s_i(\vec{p}')$. Thus if \vec{s} is a strictly proper scoring rule, traders have an incentive to change the market’s distribution to match their true beliefs. The idea is that if traders update their own beliefs over time based on market activity, the market’s distribution should eventually converge to the collective beliefs of the population.

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 \vec{p}_0 be the initial probability distribution of the market. The worst case loss of the market maker is then

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

The worst case loss of the market maker running an LMSR initialized to the uniform distribution is $b \log N$.

Note that the parameters a_1, \dots, a_N of the logarithmic scoring rule do not affect either the payoff of traders or the loss of the market maker in the LMSR. For simplicity, in the remainder of this paper when discussing the LMSR we assume that $a_i = 0$ for all i .

³In 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.

2.2 Cost Function Based Markets

As before, let $\{1, \dots, N\}$ be a set of mutually exclusive and exhaustive outcomes of an event. In a cost function based market, a market maker offers a security corresponding to each outcome i . The security associated with outcome i pays off \$1 if i happens, and \$0 otherwise.⁴

Different mechanisms can be used to determine how these securities are priced. Each mechanism is specified using a differentiable *cost function* $C : \mathbb{R}^N \rightarrow \mathbb{R}$. This cost function is simply a potential function describing the amount of money currently wagered in the market as a function of the quantity of shares purchased. If q_i is the number of shares of security i currently held by traders, and a trader would like to purchase r_i shares of each security (where r_i could be zero or even negative, representing the sale of shares), the trader must pay $C(\vec{q} + \vec{r}) - C(\vec{q})$ to the market maker. The instantaneous price of security i (that is, the price per share of an infinitely small number of shares) is then $p_i = \partial C(\vec{q}) / \partial q_i$.

We say that a cost function is *valid* if the associated prices satisfy two simple conditions:

1. For every $i \in \{1, \dots, N\}$ and every $\vec{q} \in \mathbb{R}^N$, $p_i(\vec{q}) \geq 0$.
2. For every $\vec{q} \in \mathbb{R}^N$, $\sum_{i=1}^N p_i(\vec{q}) = 1$.

The first condition ensures that the price of a security is never negative. If the current price of the security associated with an outcome i were negative, a trader could purchase shares of this security at a guaranteed profit. The second condition ensures that the prices of all securities sum to 1. If the prices summed to something less than (respectively, greater than) 1, then a trader could purchase (respectively, sell) small equal quantities of each security for a guaranteed profit. Together, these conditions ensure that there are no arbitrage opportunities within the market.

These conditions also ensure that the current prices can always be viewed as a valid probability distribution over the outcome space. In fact, these prices represent the market's current estimate of the probability that outcome i will occur.

The following theorem gives sufficient and necessary conditions for the cost function C to be valid. While these properties of cost functions have been mentioned elsewhere [5, 1], the fact that they are both sufficient and necessary for any valid cost function C is important for our later analysis. As such, we state the full proof here for completeness.

THEOREM 1. *A cost function C is valid if and only if it satisfies the following three properties:*

1. **DIFFERENTIABILITY:** *The partial derivatives $\partial C(\vec{q}) / \partial q_i$ exist for all $\vec{q} \in \mathbb{R}^N$ and $i \in \{1, \dots, N\}$.*
2. **INCREASING MONOTONICITY:** *For any \vec{q} and \vec{q}' , if $\vec{q} \geq \vec{q}'$, then $C(\vec{q}) \geq C(\vec{q}')$.*
3. **POSITIVE TRANSLATION INVARIANCE:** *For any \vec{q} and any constant k , $C(\vec{q} + k\vec{1}) = C(\vec{q}) + k$.*

PROOF. Differentiability is necessary and sufficient for the price functions to be well-defined at all points. It is easy to see that requiring the cost function to be monotonic is equivalent to requiring that $p_i(\vec{q}) \geq 0$ for all i and \vec{q} . We

⁴The dynamic parimutuel market falls outside this framework since the winning payoff depends on future trades.

will show that requiring positive translation invariance is equivalent to requiring that the prices always sum to one.

First, assume that $\sum_{i=1}^N p_i(\vec{q}) = 1$ for all \vec{q} . Fix any value of \vec{q} . Define $\vec{u} = \vec{u}(a) = \vec{q} + a\vec{1}$ and let u_i be the i th component of \vec{u} . Then for any k ,

$$\begin{aligned} C(\vec{q} + k\vec{1}) - C(\vec{q}) &= \int_0^k \frac{dC(\vec{q} + a\vec{1})}{da} da \\ &= \int_0^k \sum_{i=1}^N \frac{\partial C(\vec{u})}{\partial u_i} \frac{\partial u_i}{\partial a} da \\ &= \int_0^k \sum_{i=1}^N p_i(\vec{u}) da = k. \end{aligned}$$

This is precisely translation invariance.

Now assume instead that positive translation invariance holds. Fix any arbitrary \vec{q}' and k and define $\vec{q} = \vec{q}' + k\vec{1}$. Notice that by setting \vec{q}' and k appropriately, we can make \vec{q} take on any arbitrary values. We have,

$$\frac{\partial C(\vec{q})}{\partial k} = \sum_{i=1}^N \frac{\partial C(\vec{q})}{\partial q_i} \frac{\partial q_i}{\partial k} = \sum_{i=1}^N p_i(\vec{q}).$$

By translation invariance, $C(\vec{q}' + k\vec{1}) = C(\vec{q}') + k$. Thus,

$$\frac{\partial C(\vec{q})}{\partial k} = \frac{\partial (C(\vec{q}') + k)}{\partial k} = 1.$$

Combining the two equations, we have $\sum_{i=1}^N p_i(\vec{q}) = 1$. \square

One quantity that is useful for comparing different market mechanisms is the worst-case loss of the market maker,

$$\sup_{\vec{q} \in \mathbb{R}^N} \left(\sup_{i \in \{1, \dots, N\}} q_i - (C(\vec{q}) - C(\vec{0})) \right).$$

This is simply the difference between the maximum amount that the market maker might have to pay the winners and the amount of money collected by the market maker.

The Logarithmic Market Scoring Rule described above can be specified as a cost function based prediction market [15, 5]. The cost function of the LMSR is

$$C(\vec{q}) = b \log \sum_{i=1}^N e^{q_i/b},$$

and the corresponding prices are

$$p_i(\vec{q}) = \frac{\partial C(\vec{q})}{\partial q_i} = \frac{e^{q_i/b}}{\sum_{j=1}^N e^{q_j/b}}.$$

This formulation is equivalent to the market scoring rule formulation in the sense that a trader who changes the market probabilities from \vec{r} to \vec{r}' in the MSR formulation receives the same payoff for every outcome i as a trader who changes the quantity vectors from any \vec{q} to \vec{q}' such that $p(\vec{q}) = \vec{r}$ and $p(\vec{q}') = \vec{r}'$ in the cost function formulation.

3. LEARNING FROM EXPERT ADVICE

We now briefly review the problem of learning from expert advice. In this framework, an algorithm makes a sequence of predictions based on the advice of a set of N *experts* and receives a corresponding sequence of *losses*.⁵ The goal of the

⁵This framework could be formalized equally well in terms of *gains*, but losses are more common in the literature.

algorithm 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 $L_{\mathcal{A},T} = \sum_{t=1}^T \ell_{\mathcal{A},t} = \sum_{t=1}^T \sum_{i=1}^N w_{i,t} \ell_{i,t}$.

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 [28, 9] is an example of a no-regret algorithm. Weighted Majority uses weights

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

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 [24, 25, 20]. More specifically, the weights at time t are precisely those that minimize

$$\sum_{i=1}^N w_i L_{i,t-1} - \frac{1}{\eta} H(\vec{w})$$

among all $\vec{w} \in \Delta_N$, where H is the entropy. This makes Weighted Majority an example of broader class of algorithms collectively known as *Follow the Regularized Leader* algorithms [34, 18, 17]. This class of algorithms grew out of the following fundamental insight of Kalai and Vempala [22].

Consider first the aptly named *Follow the Leader* algorithm, which chooses weights at time t to minimize $\sum_{i=1}^N w_{i,t} L_{i,t-1}$. This algorithm simply places all of its weight on the single expert (or set of experts) with the best performance on previous examples. For this reason, this algorithm can be highly unstable, dramatically changing its weights from one time step to the next. It is easy to see that

Follow the Leader suffers $\Omega(T)$ regret in the worst case when the best expert changes frequently. For example, if there are only two experts with losses starting at $\langle 1/2, 0 \rangle$ and then alternating $\langle 0, 1 \rangle, \langle 1, 0 \rangle, \langle 0, 1 \rangle, \langle 1, 0 \rangle, \dots$, then FTL places a weight of 1 on the losing expert at every point in time.

To overcome this instability, Kalai and Vempala [22] suggested adding a random perturbation to the empirical loss of each expert, and choosing the expert that minimizes this perturbed loss.⁶ However, in general this perturbation need not be random. Instead of adding a random perturbation, it is possible to gain the necessary stability by adding a *regularizer* \mathcal{R} and choosing weights to minimize

$$\sum_{i=1}^N w_{i,t} L_{i,t-1} + \frac{1}{\eta} \mathcal{R}(\vec{w}_t). \quad (3)$$

This Follow the Regularized Leader (FTRL) approach gets around the instability of FTL and guarantees low regret for a wide variety of regularizers, as evidenced by the following bound of Hazan and Kale [18].

LEMMA 1 (HAZAN AND KALE [18]). *For any regularizer \mathcal{R} , the regret of FTRL can be bounded as*

$$\begin{aligned} L_{FTRL(\mathcal{R},\eta),T} &= \min_{i \in \{1, \dots, N\}} L_{i,T} \\ &\leq \sum_{t=1}^T \sum_{i=1}^N \ell_{i,t} (w_{i,t} - w_{i,t+1}) + \frac{1}{\eta} (\mathcal{R}(\vec{w}_T) - \mathcal{R}(\vec{w}_0)). \end{aligned}$$

This lemma quantifies the trade-off that must be considered when choosing a regularizer. If the range of the regularizer is too small, the weights will change dramatically from one round to the next, and the first term in the bound will be large. On the other hand, if the range of the regularizer is too big, the weights that are chosen will be too far from the true loss minimizers and the second term will blow up.

It is generally assumed that the regularizer \mathcal{R} is strictly convex. This assumption ensures that Equation 3 has a unique minimizer and that this minimizer can be computed efficiently. Hazan [17] shows that if \mathcal{R} is strictly convex then it is possible to achieve a regret of $O(\sqrt{T})$. In particular, by optimizing η appropriately the regret bound in Lemma 1 can be upper bounded by

$$2\sqrt{2\lambda \max_{\vec{w}, \vec{w}' \in \Delta_N} (\mathcal{R}(\vec{w}) - \mathcal{R}(\vec{w}'))T} \quad (4)$$

where $\lambda = \max_{\ell \in [0,1]^N, \vec{w} \in \Delta_N} \ell^T [\nabla^2 \mathcal{R}(\vec{w})]^{-1} \ell$.

4. INTERPRETING PREDICTION MARKETS AS NO-REGRET LEARNERS

With this foundation in place, we are ready to describe how any bounded loss market maker can be interpreted as an algorithm for learning from expert advice. The key idea is to associate *outcomes* in the market with *experts*, and *trades* made in the market with *losses* observed by the learning algorithm. We can then view the market maker as learning a probability distribution over outcomes by treating each observed trade as a training instance in the same way that the learning algorithm learns a distribution over experts.

⁶A very similar algorithm was originally developed and analyzed by Hannan in the 1950s [14].

More formally, consider any cost function based market maker with instantaneous price functions p_i for each outcome i . We convert such a market maker to an algorithm for learning from expert advice by setting the weight of expert i at time t using

$$w_{i,t} = p_i(-\epsilon \vec{L}_{t-1}), \quad (5)$$

where $\epsilon > 0$ is a tunable parameter and $\vec{L}_{t-1} = \langle L_{1,t-1}, \dots, L_{N,t-1} \rangle$ is the vector of cumulative losses at time $t-1$. In other words, the weight on expert i at time t in the learning algorithm is the instantaneous price of security i in the market when $-\epsilon L_{j,t-1}$ shares have been purchased (or $\epsilon L_{j,t-1}$ shares have been sold) of each security j . We discuss the role of the parameter ϵ in more detail below.

First note that for any valid cost function based prediction market, setting the weights as in Equation 5 entails valid expert learning algorithm. Since the prices of any valid prediction market must be non-negative and sum to one, the weights of the resulting algorithm are guaranteed to satisfy these properties too. Furthermore, the weights are a function of only the past losses of each expert, which the algorithm is permitted to observe.

Below we show that applying this conversion to any bounded-loss market maker with slowly changing prices yields a learning algorithm with $O(\sqrt{T})$ regret. The quality of the regret bound obtained depends on the trade-off between market maker loss and how quickly the prices change. We then show how this bound can be used to rederive the standard regret bound of Weighted Majority, the converse of the result of Chen et al. [6].

4.1 A Bound on Regret

In order to derive a regret bound for the learning algorithm defined in Equation 5, it is necessary to make some restrictions on how quickly the prices in the market change. If market prices change too quickly, the resulting learning algorithm will be unstable and will suffer high worst-case regret, as was the case with the naive Follow The Leader algorithm described in Section 3. To capture this idea, we introduce the notion of ϕ -stability, defined as follows.

DEFINITION 1. We say that a set of price functions \vec{p} is ϕ -stable for a constant ϕ if p_i is continuous and piecewise differentiable for all $i \in \{1, \dots, N\}$ and $\sum_{i=1}^N \sum_{j=1}^N |D_{i,j}(\vec{t})| \leq \phi$ for all \vec{t} , where

$$D_{i,j}(\vec{t}) = \begin{cases} \left. \frac{\partial p_i(\vec{q})}{\partial q_j} \right|_{\vec{q}=\vec{t}} & \text{if } \frac{\partial p_i(\vec{q})}{\partial q_j} \text{ is defined at } \vec{t}, \\ 0 & \text{otherwise.} \end{cases}$$

Defining ϕ -stability in terms of the $D_{i,j}$ allows us to quantify how slowly the prices change even when the price functions are not differentiable at all points. We can then derive a regret bound for the resulting learning algorithm using the following simple lemma, which states that when the quantity vector in the market is \vec{q} , if the price functions are ϕ -stable, then the amount of money that the market maker would collect for the purchase of a small quantity r_i of each security i is not too far from the amount that the market maker would have collected had he instead priced the shares according to the fixed price $\vec{p}(\vec{q})$. The proof is in the appendix.

LEMMA 2. Let C be any valid cost function yielding ϕ -stable prices. For any $\epsilon > 0$, any $\vec{q} \in \mathbb{R}^N$, and any $\vec{r} \in \mathbb{R}^N$

such that $|r_i| \leq \epsilon$ for $i \in \{1, \dots, N\}$,

$$\left| (C(\vec{q} + \vec{r}) - C(\vec{q})) - \sum_{i=1}^N p_i(\vec{q}) r_i \right| \leq \frac{\epsilon^2 \phi}{2}.$$

We are ready to derive the regret bound. In the following theorem, it is assumed that T is known a priori and therefore can be used to set ϵ . If it's not, a standard “doubling trick” can be applied [4]. The idea behind the doubling trick is to partition time into periods of exponentially increasing length, restarting the algorithm each period. This leads to similar bounds with only an extra factor of $\log(T)$.

THEOREM 2. Let C be any valid cost function yielding ϕ -stable prices. Let B be a bound on the worst-case loss of the market maker mechanism associated with C . Let \mathcal{A} be the expert learning algorithm with weights as in Equation 5 with $\epsilon = \sqrt{2B/(\phi T)}$. Then for any sequence of expert losses $\ell_{i,t} \in [0, 1]$ over T time steps,

$$L_{\mathcal{A},T} - \min_{i \in \{1, \dots, N\}} L_{i,T} \leq \sqrt{2B\phi T}.$$

PROOF. By setting the weights as in Equation 5, we are essentially simulating a market over N outcomes. Let $r_{i,t}$ denote the number of shares of outcome i purchased at time step t in this simulated market, and denote by \vec{r}_t the vector of these quantities for all i . Note that $r_{i,t}$ is completely in our control since we are simply simulating a market, thus we can choose to set $r_{i,t} = -\epsilon \ell_{i,t}$ for all i and t . We have that $r_{i,t} \in [-\epsilon, 0]$ for all i and t since $\ell_{i,t} \in [0, 1]$. Let $q_{i,t} = \sum_{t'=1}^t r_{i,t'}$ be the total number of outstanding shares of security i after time t , with \vec{q}_t denoting the vector over all i . The weight assigned to expert i at round t of the learning algorithm corresponds to the instantaneous price of security i in the simulated market immediately before round t , that is, $w_{i,t} = p_i(-\epsilon \vec{L}_{t-1}) = p_i(\vec{q}_{t-1})$.

By the definition of worst-case market maker loss, $\max_i q_{i,t} - (C(\vec{q}_t) - C(\vec{0})) \leq B$. It is easy to see that we can rewrite the left-hand side of this equation to obtain

$$\max_{i \in \{1, \dots, N\}} \sum_{t=1}^T r_{i,t} - \sum_{t=1}^t (C(\vec{q}_t) - C(\vec{q}_{t-1})) \leq B.$$

From Lemma 2, this gives us that

$$\max_{i \in \{1, \dots, N\}} \sum_{t=1}^T r_{i,t} - \sum_{t=1}^t \left(\sum_{i=1}^N p_i(\vec{q}_{t-1}) r_{i,t} + \frac{\epsilon^2 \phi}{2} \right) \leq B.$$

Substituting $p_i(\vec{q}_{t-1}) = w_{i,t}$ and $r_{i,t} = -\epsilon \ell_{i,t}$, we get

$$\max_{i \in \{1, \dots, N\}} \sum_{t=1}^T (-\epsilon \ell_{i,t}) - \sum_{t=1}^t \sum_{i=1}^N w_{i,t} (-\epsilon \ell_{i,t}) \leq B + \frac{\epsilon^2 \phi T}{2}$$

and so

$$\begin{aligned} L_{\mathcal{A},T} - \min_{i \in \{1, \dots, N\}} L_{i,T} &= \sum_{t=1}^T \sum_{i=1}^N w_{i,t} \ell_{i,t} - \min_i \sum_{t=1}^T \ell_{i,t} \\ &\leq \frac{B}{\epsilon} + \frac{\epsilon \phi T}{2}. \end{aligned}$$

Setting $\epsilon = \sqrt{2B/(\phi T)}$ yields the bound. \square

4.2 Rederiving the Weighted Majority Bound

Chen et al. [6] showed that the Weighted Majority regret bound can be used as a starting point to rederive the worst case loss of $b \log N$ of an LMSR market maker. Here we show that the converse is also true; by applying Theorem 2, we can rederive the Weighted Majority bound from the bounded market maker loss of LMSR. To do this, we must provide a bound on how quickly LMSR prices can change. The following lemma does just that; the proof is in the appendix.

LEMMA 3. *Let \vec{p} be the pricing function of a LMSR with parameter $b > 0$. Then*

$$\sum_{i=1}^N \sum_{j=1}^N \left| \frac{\partial p_i(\vec{q})}{\partial q_j} \right| \leq \frac{2}{b}.$$

Using Equation 5 to transform the LMSR into a learning algorithm, we end up with weights

$$w_{i,t} = \frac{e^{-\epsilon L_{i,t-1}/b}}{\sum_{j=1}^N e^{-\epsilon L_{j,t-1}/b}}.$$

Setting $\epsilon = \sqrt{2B/(\phi T)} = b\sqrt{\log N/T}$, we see that these weights are equivalent to those used by Weighted Majority with the learning rate $\eta = \epsilon/b = \sqrt{\log N/T}$. As mentioned above, this is the optimal setting of η . Notice that these weights do not depend on the value of the parameter b in the prediction market.

We can now apply Theorem 2 to rederive the standard Weighted Majority regret bound stated in Section 3. In particular, setting $B = b \log N$ and $\phi = 2/b$, we get that when $\eta = \sqrt{\log(N)/T}$,

$$L_{WM,T} - \min_{i \in \{1, \dots, N\}} L_{i,T} \leq 2\sqrt{T \log N}.$$

5. CONNECTIONS BETWEEN MARKET SCORING RULES, COST FUNCTIONS, AND REGULARIZATION

In this section, we establish the formal connections among market scoring rules, cost function based markets, and the class of Follow the Regularized Leaders algorithms. We start with a representation theorem for cost function based markets, which is crucial in our later analysis.

5.1 A Representation Theorem for Convex Cost Functions

In this section we show a representation theorem for convex cost functions. The proof of this theorem relies on the connection between convex cost functions and a class of functions known in the finance literature as convex risk measures, which was first noted by Agrawal et al. [1]. Convex risk measures were originally introduced by Föllmer and Schied [8] to model different attitudes towards risk in financial markets. A *risk measure* ρ can be viewed as a mapping from a vector of returns (corresponding to each possible outcome of an event) to a real number. The interpretation is that a vector of returns \vec{x} is “preferred to” the vector \vec{x}' under a risk measure ρ if and only if $\rho(\vec{x}) < \rho(\vec{x}')$.

Formally, a function ρ is a *convex risk measure* if it satisfies the following three properties:

1. CONVEXITY: $\rho(\vec{x})$ is a convex function of \vec{x} .

2. DECREASING MONOTONICITY: For any \vec{x} and \vec{x}' , if $\vec{x} \geq \vec{x}'$, then $\rho(\vec{x}) \leq \rho(\vec{x}')$.
3. NEGATIVE TRANSLATION INVARIANCE: For any \vec{x} and value k , $\rho(\vec{x} + k\vec{1}) = \rho(\vec{x}) - k$.

The financial interpretations of these properties are not important in our setting. More interesting for us is that Föllmer and Schied [8] provide a representation theorem that states that a function ρ is a convex risk measure if and only if it can be represented as

$$\rho(\vec{x}) = \sup_{\vec{p} \in \Delta_N} \left(- \sum_{i=1}^N p_i x_i - \alpha(\vec{p}) \right)$$

where $\alpha : \Delta_N \rightarrow (-\infty, \infty]$ is a convex, lower semi-continuous function referred to as a *penalty function*. This fact is useful because it allows us to obtain the following result, which was alluded to informally by Agrawal et al. [1]. The full proof is included here for completeness.

LEMMA 4. *A function C is a valid convex cost function if and only if it is differentiable and can be represented as*

$$C(\vec{q}) = \sup_{\vec{p} \in \Delta_N} \left(\sum_{i=1}^N p_i q_i - \alpha(\vec{p}) \right) \quad (6)$$

for a convex and lower semi-continuous function α . Furthermore, for any quantity vector \vec{q} , the price vector $\vec{p}(\vec{q})$ corresponding to C is the distribution \vec{p} maximizing $\sum_{i=1}^N p_i q_i - \alpha(\vec{p})$.

PROOF. Consider any differentiable function $C : \mathbb{R}^N \rightarrow \mathbb{R}$. Let $\rho(\vec{q}) = C(-\vec{q})$. Clearly by definition, ρ satisfies decreasing monotonicity if and only if C satisfies increasing monotonicity, and ρ satisfies negative translation invariance if and only if C satisfies positive translation invariance. Furthermore, ρ is convex if and only if C is convex. By Theorem 1, this implies that C is a valid convex cost function if and only if ρ is a convex risk measure. The first half of the lemma then follows immediately from the representation theorem of Föllmer and Schied [8] stated above.

Now, because $\alpha(\vec{p})$ is guaranteed to be convex, $\sum_{i=1}^N p_i q_i - \alpha(\vec{p})$ is a concave function of \vec{p} . The constraints $\sum_{i=1}^N p_i = 1$ and $p_i \geq 0$ define a closed convex feasible set. Thus, the problem of maximizing $\sum_{i=1}^N p_i q_i - \alpha(\vec{p})$ with respect to \vec{p} has a global optimal solution and first-order KKT conditions are both necessary and sufficient. Let $\vec{p}^*(\vec{q})$ denote an optimal \vec{p} for this optimization problem. Then, $C(\vec{q}) = \sum_{i=1}^N p_i^*(\vec{q}) q_i - \alpha(\vec{p}^*(\vec{q}))$. By the envelope theorem [29], if $C(\vec{q})$ is differentiable, we have that for any i , $p_i^*(\vec{q}) = \partial C(\vec{q}) / \partial q_i = p_i(\vec{q})$. Thus the market prices are precisely those which maximize the inner expression of the cost function. \square

Furthermore, by a version of the envelope theorem [23], to ensure that C is differentiable, it is sufficient to show that α is strictly convex and differentiable.

COROLLARY 1. *A function C is a valid convex cost function if it can be represented as in Equation 6 for a strictly convex and differentiable function α . For any \vec{q} , the price vector $\vec{p}(\vec{q})$ is the distribution \vec{p} maximizing $\sum_{i=1}^N p_i q_i - \alpha(\vec{p})$.*

The ability to represent any valid cost function in this form allows us to define a bound on the worst-case loss of the market maker in terms of the penalty function of the corresponding convex risk measure.

LEMMA 5. *The worst-case loss of the market maker defined by the cost function in Equation 6 is no more than*

$$\sup_{\vec{p}, \vec{p}' \in \Delta_N} (\alpha(\vec{p}) - \alpha(\vec{p}')).$$

PROOF. The worst-case loss of the market maker is

$$\begin{aligned} & \sup_{\vec{q} \in \mathbb{R}^N} \left(\max_{i \in \{1, \dots, N\}} q_i - C(\vec{q}) \right) + C(\vec{0}) \\ &= \sup_{\vec{q} \in \mathbb{R}^N} \left(\max_{i \in \{1, \dots, N\}} q_i - \sup_{\vec{p} \in \Delta_N} \left(\sum_{i=1}^N p_i q_i - \alpha(\vec{p}) \right) \right) \\ & \quad + \sup_{\vec{p}' \in \Delta_N} (-\alpha(\vec{p}')) \\ &\leq \sup_{\vec{q} \in \mathbb{R}^N} \left(\max_{i \in \{1, \dots, N\}} q_i - \left(\sup_{\vec{p} \in \Delta_N} \sum_{i=1}^N p_i q_i - \sup_{\vec{p} \in \Delta_N} (\alpha(\vec{p})) \right) \right) \\ & \quad + \sup_{\vec{p}' \in \Delta_N} (-\alpha(\vec{p}')) \\ &= \sup_{\vec{q} \in \mathbb{R}^N} \left(\max_{i \in \{1, \dots, N\}} q_i - \max_{i \in \{1, \dots, N\}} q_i \right) + \sup_{\vec{p} \in \Delta_N} (\alpha(\vec{p})) \\ & \quad + \sup_{\vec{p}' \in \Delta_N} (-\alpha(\vec{p}')) \\ &= \sup_{\vec{p}, \vec{p}' \in \Delta_N} (\alpha(\vec{p}) - \alpha(\vec{p}')). \end{aligned}$$

The inequality follows from the fact that for any functions f and g over any domain \mathcal{X} , $\sup_{x \in \mathcal{X}} (f(x) - g(x)) \geq \sup_{x \in \mathcal{X}} f(x) - \sup_{x' \in \mathcal{X}} g(x')$. \square

5.2 Relationship Between Convex Cost Functions and Market Scoring Rules

As described in Section 2, the Logarithmic Market Scoring Rule market maker can be defined as either a market scoring rule or a cost function based market. The LMSR is not unique in this regard. As we show in this section, any regular, strictly proper market scoring rule with differentiable scoring functions can be represented as a cost function based market. Likewise, any convex cost function satisfying a few mild conditions corresponds to a market scoring rule. As long as the market probabilities are nonzero, the market scoring rule and corresponding cost function based market are equivalent. More precisely, a trader who changes the market probabilities from \vec{r} to \vec{r}' in the market scoring rule is guaranteed to receive the same payoff for every outcome i as a trader who changes the quantity vectors from any \vec{q} to \vec{q}' such that $p(\vec{q}) = \vec{r}$ and $p(\vec{q}') = \vec{r}'$ in the cost function formulation as long as every component of \vec{r} and \vec{r}' is nonzero. Moreover, any price vector that is achievable in the market scoring rule (that is, any \vec{p} for which $s_i(\vec{p})$ is finite for all i) is achievable by the 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 [5]. They pointed out that the MSR with scoring function \vec{s} and the cost function based market with cost function C are equivalent if for all \vec{q} and all outcomes i , $C(\vec{q}) = q_i - s_i(\vec{p})$. However, they

did not provide any guarantees about the circumstances under which this condition can be satisfied. Agrawal et al. [1] also made use of the equivalence between markets when this strong condition holds. Our result gives very general precise conditions under which an MSR is equivalent to a cost function based market.

Recall from Lemma 4 that any convex cost function C can be represented as $C(\vec{q}) = \sup_{\vec{p} \in \Delta_N} \left(\sum_{i=1}^N p_i q_i - \alpha(\vec{p}) \right)$ for a convex function α . Let α_C denote the function α corresponding to the cost function C . In the following, we consider cost functions derived from scoring rules \vec{s} by setting

$$\alpha_C(\vec{p}) = \sum_{i=1}^N p_i s_i(\vec{p}) \quad (7)$$

and scoring rules derived from convex cost functions with

$$s_i(\vec{p}) = \alpha_C(\vec{p}) - \sum_{j=1}^N \frac{\partial \alpha_C(\vec{p})}{\partial p_j} p_j + \frac{\partial \alpha_C(\vec{p})}{\partial p_i}. \quad (8)$$

We show that there is a mapping between a mildly restricted class of convex cost function based markets and a mildly restricted class of strictly proper market scoring rules such that for every pair in the mapping, Equations 7 and 8 both hold. Furthermore, we show that the markets satisfying these equations are equivalent in the sense described above.

THEOREM 3. *There is a one-to-one and onto mapping between the set of convex cost function based markets with strictly convex and differentiable potential functions α_C and the class of strictly proper, regular market scoring rules with differentiable scoring functions \vec{s} such that for each pair in the mapping, Equations 7 and 8 hold.*

Furthermore, each pair of markets in this mapping are equivalent when prices for all outcomes are positive, that is, 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. Additionally, every price vector \vec{p} achievable in the market scoring rule is achievable in the cost function based market.

PROOF. We first show that the function α_C in Equation 7 is strictly convex and differentiable and the scoring rule in Equation 8 is regular, strictly proper and differentiable. We then show that Equations 7 and 8 are equivalent. Finally, we show the equivalence between the two markets.

Consider the function α_C in Equation 7. Since we have assumed that s_i is differentiable for all i , α_C is differentiable too. Additionally, it is known that a scoring rule is strictly proper only if its expected value is strictly convex [12], so α_C is strictly convex.

Consider the scoring rule defined in Equation 8. By Theorem 1 of Gneiting and Raftery [12], a regular scoring rule $s_i(\vec{p})$ is strictly proper if and only if there exists a strictly convex function $G(\vec{p})$ such that

$$s_i(\vec{p}) = G(\vec{p}) - \sum_{j=1}^N p_j \dot{G}_j(\vec{p}) + \dot{G}_i(\vec{p}), \quad (9)$$

where $\dot{G}_j(\vec{p})$ is any subderivative of G with respect to p_j (if G is differentiable, $\dot{G}_j = \partial G(\vec{p}) / \partial p_j$). This immediately implies that the scoring rule defined in Equation 8 is a regular

strictly proper scoring rule since $\alpha(\vec{p})$ is strictly convex. We will see below that s_i is also differentiable.

It is easy to see that Equation 8 implies Equation 7. Suppose Equation 8 holds. Then

$$\begin{aligned} \sum_{i=1}^N p_i s_i(\vec{p}) &= \sum_{i=1}^N p_i \left(\alpha_C(\vec{p}) - \sum_{j=1}^N \frac{\partial \alpha_C(\vec{p})}{\partial p_j} p_j + \frac{\partial \alpha_C(\vec{p})}{\partial p_i} \right) \\ &= \alpha_C(\vec{p}). \end{aligned}$$

This also shows that s_i is differentiable for all i , since the derivative of α_C is well-defined at all points and

$$\frac{\partial \alpha_C(\vec{p})}{\partial p_i} = s_i(\vec{p}) + \sum_{i=1}^N \frac{\partial s_i(\vec{p})}{\partial p_i}.$$

To see that Equation 7 implies Equation 8, suppose that Equation 7 holds. We know that the scoring rule \vec{s} can be expressed as in Equation 9 for some function G . For this particular G ,

$$\alpha_C(\vec{p}) = \sum_{i=1}^N p_i \left(G(\vec{p}) - \sum_{j=1}^N p_j \dot{G}_j(\vec{p}) + \dot{G}_i(\vec{p}) \right) = G(\vec{p}).$$

Since $G(\vec{p}) = \alpha_C(\vec{p})$ and α_C is differentiable (meaning that $\partial \alpha_C / \partial p_i$ is the only subderivative of α_C with respect to p_i), this implies Equation 8.

We have established the equivalence between Equations 7 and 8. We now show that a trader gets exactly the same profit for any realized outcome in the two markets if the market prices are positive.

Suppose in the cost function based market a trader changes the outstanding shares from \vec{q} to \vec{q}' . This trade changes the market price from $\vec{p}(\vec{q})$ to $\vec{p}(\vec{q}')$. If outcome i occurs, the trader's profit is

$$\begin{aligned} &(q'_i - q_i) - (C(\vec{q}') - C(\vec{q})) \\ &= (q'_i - q_i) - \left(\sum_{j=1}^N p_j(\vec{q}') q'_j - \alpha_C(\vec{p}(\vec{q}')) \right) \\ &\quad + \left(\sum_{j=1}^N p_j(\vec{q}) q_j - \alpha_C(\vec{p}(\vec{q})) \right) \\ &= \left(q'_i - \sum_{j=1}^N p_j(\vec{q}') q'_j + \alpha_C(\vec{p}(\vec{q}')) \right) \\ &\quad - \left(q_i - \sum_{j=1}^N p_j(\vec{q}) q_j + \alpha_C(\vec{p}(\vec{q})) \right). \end{aligned} \quad (10)$$

From Lemma 4, we know that $\vec{p}(\vec{q})$ is the optimal solution to the convex optimization $\max_{\vec{p} \in \Delta_N} (\sum_{i=1}^N p_i q_i - \alpha_C(\vec{p}))$. The Lagrange function of this optimization problem is

$$L = \left(\sum_{i=1}^N p_i q_i - \alpha_C(\vec{p}) \right) - \lambda \left(\sum_{i=1}^N p_i - 1 \right) + \sum_{i=1}^N \mu_i p_i.$$

Since $\vec{p}(\vec{q})$ is optimal, the KKT conditions require that $\partial L / \partial p_i = 0$, which implies that for all i ,

$$q_i = \frac{\partial \alpha_C(\vec{p}(\vec{q}))}{\partial p_i(\vec{q})} + \lambda(\vec{q}) - \mu_i(\vec{q}), \quad (11)$$

where $\mu_i(\vec{q}) \geq 0$ and $\mu_i(\vec{q}) p_i(\vec{q}) = 0$. Plugging (11) into (10), we have

$$\begin{aligned} &(q'_i - q_i) - (C(\vec{q}') - C(\vec{q})) \\ &= \left(\frac{\partial \alpha_C(\vec{p}(\vec{q}'))}{\partial p_i(\vec{q}')} - \sum_{j=1}^N p_j(\vec{q}') \frac{\partial \alpha_C(\vec{p}(\vec{q}'))}{\partial p_j(\vec{q}')} + \alpha_C(\vec{p}(\vec{q}')) - \mu_i(\vec{q}') \right) \\ &\quad - \left(\frac{\partial \alpha_C(\vec{p}(\vec{q}))}{\partial p_i(\vec{q})} - \sum_{j=1}^N p_j(\vec{q}) \frac{\partial \alpha_C(\vec{p}(\vec{q}))}{\partial p_j(\vec{q})} + \alpha_C(\vec{p}(\vec{q})) - \mu_i(\vec{q}) \right) \\ &= (s_i(\vec{p}(\vec{q}')) - \mu_i(\vec{q}')) - (s_i(\vec{p}(\vec{q})) - \mu_i(\vec{q})). \end{aligned} \quad (12)$$

When $p_i(\vec{q}) > 0$ and $p_i(\vec{q}') > 0$, $\mu_i(\vec{q}) = \mu_i(\vec{q}') = 0$. In this case, the profit of the trader in the cost function based market is the same as that in the market scoring rule market when he changes the market probability from $\vec{p}(\vec{q})$ to $\vec{p}(\vec{q}')$.

Finally, observe that using the cost function based market it is possible to achieve any price vector \vec{r} with finite scores $s_i(\vec{r})$ by setting $q_i = s_i(\vec{r})$ for all i . By Lemma 4, for this setting of \vec{q} , $\vec{p}(\vec{q})$ is the vector \vec{p} that maximizes $\sum_{i=1}^N p_i s_i(\vec{r}) - \sum_{i=1}^N p_i s_i(\vec{p})$. Since \vec{s} is strictly proper, this is maximized at $\vec{p} = \vec{r}$. Since \vec{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 \vec{p} on the exterior as long as $s_i(\vec{p})$ is finite for all i). \square

Cost function based market makers are more natural for implementation purposes. The correspondence between market scoring rules and cost function based market makers, $C(\vec{q}) = q_i - s_i(\vec{p})$, noted by Chen and Pennock [5] does not provide a general way to find the cost function given a market scoring rule because \vec{p} is the vector of partial derivatives of $C(\vec{q})$. Theorem 3 for the first time allows us to translate any strictly proper, regular market scoring rule into its cost function formulation. The cost function is represented as a convex optimization which can be easily solved with standard methods given any \vec{q} .

5.3 Convex Cost Functions and FTRL

Consider a prediction market with a convex cost function represented as $C(\vec{q}) = \sup_{\vec{p} \in \Delta_N} \left(\sum_{i=1}^N p_i q_i - \alpha(\vec{p}) \right)$ and the corresponding learning algorithm with weights $w_{i,t} = p_i(-\epsilon \vec{L}_{t-1})$. (Recall that $\vec{L}_{t-1} = \langle L_{1,t-1}, \dots, L_{N,t-1} \rangle$ is the vector of cumulative losses at time $t-1$.) By Lemma 4, the weights chosen at time t are those that maximize the expression $-\epsilon \sum_{i=1}^N w_i L_{i,t-1} - \alpha(\vec{w})$, or equivalently, those that minimize the expression

$$\sum_{i=1}^N w_i L_{i,t-1} + \frac{1}{\epsilon} \alpha(\vec{w}).$$

This expression is of precisely the same form as Equation 3, with α playing the role of the regularizer and ϵ controlling the trade-off between the regularizer and the empirical loss. This implies that every convex cost function based prediction market can be interpreted as a Follow the Regularized Leader algorithm with a convex regularizer! By applying Theorem 2 and Lemma 5, we can easily bound the regret of the resulting algorithm as follows.

THEOREM 4. *Let C be any valid convex cost function yielding ϕ -stable prices, and let α_C be the penalty function associated with C . Let \mathcal{A} be the expert learning algorithm with weights as in Equation 5 with $\epsilon =$*

$\sqrt{2 \sup_{\vec{p}, \vec{p}' \in \Delta_N} (\alpha_C(\vec{p}) - \alpha_C(\vec{p}') / (\phi T))}$. Then for any sequence of expert losses $\ell_{i,t} \in [0, 1]$ over T time steps,

$$L_{\mathcal{A}, T} - \min_{i \in \{1, \dots, N\}} L_{i, T} \leq \sqrt{2T\phi \sup_{\vec{p}, \vec{p}' \in \Delta_N} (\alpha_C(\vec{p}) - \alpha_C(\vec{p}'))}.$$

This bound is very similar to the bound for FTRL given in Equation 4, with ϕ playing the role of λ .

The connections we established in the previous section imply that every strictly proper market scoring rule can also be interpreted as a FTRL algorithm, now with a *strictly* convex regularizer. Conversely, any FTRL algorithm with a differentiable and strictly convex regularizer can be viewed as choosing weights at time t to minimize the quantity

$$\sum_{i=1}^N w_i (\epsilon L_{i,t-1} + s_i(\vec{w}))$$

for a strictly proper scoring rule \vec{s} . Perhaps it is no surprise that the weight updates of FTRL algorithms can be framed in terms of proper scoring rules given that proper scoring rules are commonly used as loss functions in machine learning [3, 32] and FTRL has previously been connected to Bregman divergences [34, 18, 17] which are known to be related to scoring rules [12].

This connection hints at why market scoring rules and convex cost function based markets may be able to obtain accurate estimates of probability distributions in practice. Both types of markets are essentially *learning* the distributions by treating market trades as training data. Beyond that, both markets correspond to well-understood learning algorithms with stable weights and guarantees of no regret.

5.4 Relation to the SCPM

Agrawal et al. [1] present another way of describing convex cost function based prediction markets, which they call the Sequential Convex Pari-Mutuel Mechanism (SCPM). The SCPM is defined in terms of limit orders instead of market prices, but the underlying mathematics are essentially the same. In the SCPM, traders specify a maximum quantity of shares that they would like to purchase and a maximum price per share that they are willing to spend. The market then decides how many shares of the trade to accept by solving a convex optimization problem.

Agrawal et al. show that for every SCPM, there is an equivalent convex cost function based market. For each limit order, the number of shares accepted by the market maker in the SCPM is the minimum of the number of shares requested by the trader and the number of shares that it would take to drive the market price of the shares in the corresponding cost function based market to the limit price of the trader. Thus our results imply that any SCPM mechanism can be interpreted as a FTRL algorithm.

We remark that Agrawal et al. [1] also describe an interpretation of the SCPM in terms of convex risk measures and suggest that the associated penalty function is related to the underlying problem of learning the distribution over outcomes. However, their interpretation is very different from ours. They view the penalty function as characterizing “the market maker’s commitment to learning the true distribution” since it impacts both the worst case market maker loss and the willingness of the market maker to accept limit

orders. On the contrary, we view the penalty function as a regularizer necessary to make the market prices stable.

6. EXAMPLE: THE QUADRATIC MSR AND ONLINE GRADIENT DESCENT

In the previous section we described the relationship between market scoring rules, cost function based markets with convex cost functions, and Follow the Regularized Leader algorithms. We discussed how the Logarithmic Market Scoring Rule can be represented equivalently as a cost function based market, and how it corresponds to Weighted Majority in the expert learning setting. In this section, we illustrate the relationship through another example. In particular, we show that the Quadratic Market Scoring Rule can be written equivalently as a cost function based market (namely the Quad-SCPM of Agrawal et al. [1]). We then show that this market corresponds to the well-studied online gradient descent algorithm in the learning setting and give a bound on the regret of this algorithm using Theorem 4.

The Quadratic Market Scoring Rule (QMSR) is the market scoring rule corresponding to the quadratic scoring function in Equation 1. As was the case in the LMSR, the parameters a_1, \dots, a_N do not affect the prices or payments of this market. As such, we assume that $a_i = 0$ for all i .

Theorem 3 implies that we can construct a cost function based market with equivalent payoffs to the QMSR whenever prices are nonzero using the cost function

$$\begin{aligned} C(\vec{q}) &= \sup_{\vec{p} \in \Delta_N} \left(\sum_{i=1}^N p_i q_i - \sum_{i=1}^N p_i b \left(2p_i - \sum_{i=1}^N p_i^2 \right) \right) \\ &= \sup_{\vec{p} \in \Delta_N} \left(\sum_{i=1}^N p_i q_i - b \sum_{i=1}^N p_i^2 \right). \end{aligned}$$

This is precisely the cost function associated with the Quad-SCPM market with a uniform prior, which was previously known to be equivalent to the QMSR when prices are nonzero [1]. The worst case loss of the market maker in both markets is $b(N-1)/N$.

Following the argument in Section 5.3, this market corresponds to the FTRL algorithm with regularizer $\eta = 1/b$ and $\mathcal{R}(\vec{w}) = \sum_{i=1}^N w_i^2$. It has been observed that using FTRL with a regularizer of this form is equivalent to online gradient descent [19, 17]. Thus we can use Theorem 4 to show a regret bound for gradient descent.

We first show that the Quad-SCPM prices are ϕ -stable for $\phi = (N^2 - 1)/(2b) < N^2/(2b)$. (See the appendix for details.) We can therefore apply Theorem 4 using $\phi = N^2/(2b)$ and $\sup_{\vec{p}, \vec{p}' \in \Delta_N} (\alpha(\vec{p}) - \alpha(\vec{p}')) = b(N-1)/N < b$ to see that for gradient descent,

$$L_{GD, T} - \min_{i \in \{1, \dots, N\}} L_{i, T} \leq N\sqrt{T}.$$

This matches the known regret bound for general gradient descent applied to the experts setting [35].

7. DISCUSSION

We have demonstrated the elegant mathematical connection between market scoring rules, cost function based prediction markets, and no-regret learning. This connection is thought-provoking on its own, as it yields to new interpretations of well-known prediction market mechanisms. The

interpretation of the penalty function as a regularizer can shed some light on which market scoring rule or cost function based market is best to run under different assumptions about traders. We are hopeful that future advancements in either no-regret learning or prediction market design will inspire advancements in the other field as well. For example, the discovery of a new well-behaved FTRL algorithm with nice properties would suggest that the corresponding market maker mechanism might have nice properties too.

Additionally, this connection has the potential to be of use in the design of new prediction market mechanisms and learning algorithms. In recent years there has been a great interest in finding ways to tractably run market scoring rules over combinatorial or infinite outcome spaces [31, 7, 10, 6]. For example, a market maker might wish to accept bets over permutations (“horse A will finish the race ahead of horse B”), Boolean spaces (“either a Democrat will win the 2010 senate race in Delaware or a Democrat will win in North Dakota”), or real numbers (“Google’s revenue in the first quarter of 2010 will be between $\$x$ and $\$y$ ”), in which case simply running a naive implementation of an LMSR (for example) would be infeasible. As mentioned above, by exploiting the connection between Weighted Majority and the LMSR, Chen et al. [6] showed that an extension of the Weighted Majority algorithm to permutation learning [20] could be used to approximate prices in an LMSR over permutations. Given our new understanding of the connection between markets and learning and the growing literature on no-regret algorithms for large or infinite sets of experts [21], it seems likely that similar learning-based techniques could be developed to calculate market prices for other types of large outcome spaces too.

Acknowledgments

This material is based upon work supported by NSF under CNS-0937060 to the CRA for the CIFellows Project, CCF-0953516, and CCF-0915016. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors alone.

References

- [1] S. Agrawal, E. Delage, M. Peters, Z. Wang, and Y. Ye. A unified framework for dynamic pari-mutuel information market design. In *ACM EC*, 2009.
- [2] G. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950.
- [3] A. Buja, W. Stuetzle, and Y. Shen. Loss functions for binary class probability estimation and classification: Structure and applications. Working draft, November 2005.
- [4] N. Cesa-Bianchi, Y. Freund, D. P. Helmbold, D. Haussler, R. Schapire, and M. Warmuth. How to use expert advice. *Journal of the ACM*, 44(3):427–485, 1997.
- [5] Y. Chen and D. M. Pennock. A utility framework for bounded-loss market makers. In *UAI*, 2007.
- [6] Y. Chen, L. Fortnow, N. Lambert, D. M. Pennock, and J. Wortman. Complexity of combinatorial market makers. In *ACM EC*, 2008.
- [7] Y. Chen, S. Goel, and D. M. Pennock. Pricing combinatorial markets for tournaments. In *ACM STOC*, 2008.
- [8] H. Föllmer and A. Schied. Convex measures of risk and trading constraints. *Finance and Stochastics*, 6(4), 2002.
- [9] 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.
- [10] X. Gao, Y. Chen, and D. M. Pennock. Betting on the real line. In *WINE*, 2009.
- [11] 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.
- [12] T. Gneiting and A. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.
- [13] I. J. Good. Rational decisions. *Journal of the Royal Statistical Society, Series B (Methodological)*, 14(1):107–114, 1952.
- [14] J. Hannan. Approximation to Bayes risk in repeated play. In M. Dresher, A. W. Tucker, and P. Wolfe, editors, *Contributions to the Theory of Games, Volume III*, pages 97–139. Princeton University Press, Princeton, 1957.
- [15] R. Hanson. Combinatorial information market design. *Information Systems Frontiers*, 5(1):105–119, 2003.
- [16] R. Hanson. Logarithmic market scoring rules for modular combinatorial information aggregation. *Journal of Prediction Markets*, 1(1):3–15, 2007.
- [17] E. Hazan. A survey: The convex optimization approach to regret minimization. Working draft, September 2009.
- [18] E. Hazan and S. Kale. Extracting certainty from uncertainty: Regret bounded by variation in costs. In *COLT*, 2008.
- [19] E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2–3):169–192, 2007.
- [20] D. Helmbold and M. Warmuth. Learning permutations with exponential weights. *JMLR*, 10:1705–1736, 2009.
- [21] M. Hutter and J. Poland. Adaptive online prediction by following the perturbed leader. *JMLR*, 6:639–660, 2005.
- [22] A. T. Kalai and S. Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.
- [23] T. Kim. Differentiability of the value function: A new characterization. *Seoul Journal of Economics*, 6(3), 1993.
- [24] J. Kivinen and M. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Journal of Information and Computation*, 132(1):1–63, 1997.
- [25] J. Kivinen and M. Warmuth. Averaging expert predictions. In *EuroCOLT*, 1999.
- [26] N. Lambert, D. M. Pennock, and Y. Shoham. Eliciting properties of probability distributions. In *ACM EC*, 2008.
- [27] J. Ledyard, R. Hanson, and T. Ishikida. An experimental test of combinatorial information markets. *Journal of Economic Behavior and Organization*, 69:182–189, 2009.
- [28] N. Littlestone and M. Warmuth. The weighted majority algorithm. *Info. and Computation*, 108(2):212–261, 1994.
- [29] P. Milgrom and I. Segal. Envelope theorems for arbitrary choice sets. *Econometrica*, 70(2):583–601, 2002.
- [30] D. M. Pennock. A dynamic pari-mutuel market for hedging, wagering, and information aggregation. In *ACM EC*, 2004.
- [31] 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.
- [32] M. D. Reid and R. C. Williamson. Surrogate regret bounds for proper losses. In *ICML*, 2009.
- [33] L. J. Savage. Elicitation of personal probabilities and expectations. *Journal of the American Statistical Association*, 66(336):783–801, 1971.
- [34] S. Shalev-Shwartz and Y. Singer. A primal-dual perspective of online learning algorithms. *Machine Learning*, 69(2–3):115–142, 2007.
- [35] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, 2003.

APPENDIX

A. PROOF OF LEMMA 2

Fix the vectors \vec{q} and \vec{r} . Let $\vec{u}(s) = \vec{q} + s\vec{r}$. Similarly to how we have defined $D_{i,j}(\vec{t})$, define

$$D_i(x) = \begin{cases} \left. \frac{\partial p_i(\vec{u}(s))}{\partial s} \right|_{s=x} & \text{if } \frac{\partial p_i(\vec{u}(s))}{\partial s} \text{ is defined at } x, \\ 0 & \text{otherwise.} \end{cases}$$

Using D_i in place of the derivative allows us to integrate over the derivative even when it is not defined at single points. For any point at which the derivatives are defined, we have

$$\frac{\partial C(\vec{u}(s))}{\partial s} = \sum_{i=1}^N \frac{\partial C(\vec{u}(s))}{\partial u_i(s)} \frac{\partial u_i(s)}{\partial s} = \sum_{i=1}^N p_i(\vec{u}(\cdot)) r_i.$$

Applying the fundamental theorem of calculus, we have that

$$\begin{aligned} C(\vec{q} + \vec{r}) - C(\vec{q}) &= \int_0^1 \left. \frac{\partial C(\vec{u}(s))}{\partial s} \right|_{s=x} dx = \int_0^1 \sum_{i=1}^N p_i(\vec{u}(x)) r_i dx \\ &= \int_0^1 \sum_{i=1}^N \left(p_i(\vec{u}(0)) + \int_0^x D_i(y) dy \right) r_i dx \\ &= \sum_{i=1}^N p_i(\vec{q}) r_i + \int_0^1 \int_0^x \sum_{i=1}^N r_i D_i(y) dy dx. \end{aligned}$$

Rearranging terms, this gives us that

$$C(\vec{q} + \vec{r}) - C(\vec{q}) - \sum_{i=1}^N p_i(\vec{q}) r_i = \int_0^1 \int_0^x \sum_{i=1}^N r_i D_i(y) dy dx.$$

To prove the lemma, it is sufficient to bound the absolute value of the expression on the right. This is where the ϕ -stability of the prices comes into play. At any point where the derivatives are defined,

$$\begin{aligned} \sum_{i=1}^N \left| \frac{\partial p_i(\vec{u}(s))}{\partial s} \right| &= \sum_{i=1}^N \left| \sum_{j=1}^N \frac{\partial p_i(\vec{q})}{\partial q_j} r_j \right| \leq \epsilon \sum_{i=1}^N \sum_{j=1}^N \left| \frac{\partial p_i(\vec{q})}{\partial q_j} \right| \\ &\leq \epsilon \phi. \end{aligned}$$

Since we have assumed that the prices are piecewise differentiable this implies that

$$\begin{aligned} \left| \int_0^1 \int_0^x \sum_{i=1}^N r_i D_i(y) dy dx \right| &\leq \epsilon \int_0^1 \int_0^x \sum_{i=1}^N |D_i(y)| dy dx \\ &\leq \epsilon \int_0^1 \int_0^x \epsilon \phi dy dx = \epsilon^2 \phi \int_0^1 x dx = \frac{\epsilon^2 \phi}{2}. \end{aligned}$$

This bounds the absolute value of the right hand side of the equation above and proves the lemma. \square

B. PROOF OF LEMMA 3

For every i and every $j \neq i$, we have

$$\begin{aligned} \frac{\partial p_i(\vec{q})}{\partial q_i} &= \frac{\partial}{\partial q_i} \frac{e^{q_i/b}}{\sum_{j=1}^N e^{q_j/b}} \\ &= \frac{1}{b} \frac{e^{q_i/b} \sum_{j=1}^N e^{q_j/b} - (e^{q_i/b})^2}{\left(\sum_{j=1}^N e^{q_j/b} \right)^2} \\ &= \frac{1}{b} (p_i(\vec{q}) - p_i(\vec{q})^2) = \frac{1}{b} p_i(\vec{q}) \sum_{j \neq i} p_j(\vec{q}) \end{aligned}$$

and

$$\begin{aligned} \frac{\partial p_i(\vec{q})}{\partial q_j} &= \frac{\partial}{\partial q_j} \frac{e^{q_i/b}}{\sum_{j=1}^N e^{q_j/b}} = \frac{1}{b} \frac{-e^{q_i/b} e^{q_j/b}}{\left(\sum_{j=1}^N e^{q_j/b} \right)^2} \\ &= -\frac{1}{b} p_i(\vec{q}) p_j(\vec{q}). \end{aligned}$$

Thus we have

$$\begin{aligned} \sum_{i=1}^N \sum_{j=1}^N \left| \frac{\partial p_i(\vec{q})}{\partial q_j} \right| &= \sum_{i=1}^N \left| \frac{\partial p_i(\vec{q})}{\partial q_i} \right| + \sum_{i=1}^N \sum_{j \neq i} \left| \frac{\partial p_i(\vec{q})}{\partial q_j} \right| \\ &= \frac{2}{b} \sum_{i=1}^N \sum_{j \neq i} p_i(\vec{q}) p_j(\vec{q}). \end{aligned}$$

We would like to find the prices that maximize this quantity. Dropping the argument \vec{q} to simply notation, this is equivalent to solving a simple optimization problem:

$$\text{maximize}_{\vec{p} \in \Delta_N} \frac{2}{b} \sum_{i=1}^N \sum_{j \neq i} p_i p_j.$$

It is straight-forward to show (e.g., using the KKT conditions) that this expression is maximized when the prices are equal across all securities, so $p_i = 1/N$ for all i . Then

$$\sum_{i=1}^N \sum_{j=1}^N \left| \frac{\partial p_i(\vec{q})}{\partial q_j} \right| \leq \frac{2}{b} \frac{N(N-1)}{N^2} \leq \frac{2}{b}.$$

\square

C. STABILITY OF QUAD-SCPM PRICES

The cost function of Quad-SCPM can be written as

$$C(\vec{q}) = \sup_{\vec{p} \in \Delta_N} \left(\sum_{i=1}^N p_i q_i - b \sum_{i=1}^N p_i^2 \right).$$

By Lemma 4, the price function of Quad-SCPM is defined by the optimal solution to the optimization problem in this function. The Lagrange function corresponding to the constrained optimization problem is

$$L = \left(\sum_{i=1}^n p_i q_i - b \sum_{i=1}^N p_i^2 \right) - \lambda \left(\sum_{i=1}^N p_i - 1 \right) + \sum_{i=1}^N \mu_i p_i.$$

Because the objective function is strictly concave in \vec{p} , there is a unique optimal solution that satisfies the KKT conditions: $\partial L / \partial p_i = 0 \forall i$, $\sum_{i=1}^N p_i = 1$, $p_i \geq 0 \forall i$, $\mu_i \geq 0 \forall i$,

and $\mu_i p_i = 0 \forall i$. Thus, the price function is defined by

$$\begin{cases} p_i = \frac{1}{N} + \frac{q_i + \mu_i}{2b} - \frac{\sum_{j=1}^N (q_j + \mu_j)}{2bN} \forall i \\ p_i \mu_i = 0 \forall i \\ p_i \geq 0 \forall i \\ \mu_i \geq 0 \forall i. \end{cases} \quad (13)$$

When $1/N + q_i/(2b) - \sum_{j=1}^N q_j/(2bN) > 0$, $\mu_i = 0$ and $p_i > 0$ for all i and the price function is

$$p_i(\vec{q}) = \frac{1}{N} + \frac{q_i}{2b} - \frac{\sum_{j=1}^N q_j}{2bN},$$

which is the same as that of QMSR. In this case, we have

$$\frac{\partial p_i}{\partial q_i} = \frac{1}{2b} - \frac{1}{2bN} \forall i \quad \text{and} \quad \frac{\partial p_i}{\partial q_j} = -\frac{1}{2bN} \forall j \neq i.$$

Consider the case in which the prices for some outcomes equal 0. Given \vec{q} , let $M = \{m : p_m > 0, \mu_m = 0\}$ be the set of outcomes that have positive prices. Let $K = \{k : p_k = 0, \mu_k > 0\}$ be the set of the outcomes that have a positive μ_k . Let $L = \{l : p_l = 0, \mu_l = 0\}$ be the set of outcomes for which both p_l and μ_l are 0. Denote $\tilde{p} = \min_{m \in M} p_m$ and $\tilde{\mu} = \min_{k \in K} \mu_k$. For $0 < |\epsilon| < \min\{\tilde{\mu}, 2b\tilde{p}(N - |K| - |L|)\}$, we consider the following cases:

- $L = \emptyset$ and $i \in M$

Consider changing \vec{q} to \vec{q}' , where $q'_i = q_i + \epsilon$ and $q'_j = q_j \forall j \neq i$. We can verify that the price function for \vec{q}' is defined by

$$\begin{cases} p'_i = p_i + (\frac{1}{2b} - \frac{1}{2b(N-|K|)})\epsilon \\ p'_j = p_j - \frac{\epsilon}{2b(N-|K|)} \quad \forall j \in M, j \neq i \\ p'_k = p_k = 0 \quad \forall k \in K \\ \mu'_j = \mu_j = 0 \quad \forall j \in M \\ \mu'_k = \mu_k + \frac{\epsilon}{N-|K|} \quad \forall k \in K \end{cases}$$

where \vec{p} and $\vec{\mu}$ are the prices and Lagrange multipliers for \vec{q} . Hence, by the definition of derivatives, we have

$$\frac{\partial p_i}{\partial q_i} = \frac{1}{2b} - \frac{1}{2b(N-|K|)} \quad \forall i \in M,$$

$$\frac{\partial p_j}{\partial q_i} = -\frac{1}{2b(N-|K|)} \quad \forall i, j \in M, i \neq j,$$

$$\frac{\partial p_k}{\partial q_i} = 0 \quad \forall k \in K, i \in M.$$

- $L \neq \emptyset$ and $i \in M$

Consider changing \vec{q} to \vec{q}' , where $q'_i = q_i + \epsilon$ and $q'_j = q_j \forall j \neq i$. The new prices are defined by

$$\begin{cases} p'_i = p_i + (\frac{1}{2b} - \frac{1}{2b(N-|K|-|L|)})\epsilon \\ p'_j = p_j - \frac{\epsilon}{2b(N-|K|-|L|)} \quad \forall j \in M, j \neq i \\ p'_k = p_k = 0 \quad \forall k \in K \\ p'_l = p_l = 0 \quad \forall l \in L \\ \mu'_j = \mu_j = 0 \quad \forall j \in M \\ \mu'_k = \mu_k + \frac{\epsilon}{N-|K|-|L|} \quad \forall k \in K \\ \mu'_l = \mu_l + \frac{\epsilon}{N-|K|-|L|} \quad \forall l \in L \end{cases}$$

if $\epsilon > 0$, and

$$\begin{cases} p'_i = p_i + (\frac{1}{2b} - \frac{1}{2b(N-|K|)})\epsilon \\ p'_j = p_j - \frac{\epsilon}{2b(N-|K|)} \quad \forall j \in M, j \neq i \\ p'_k = p_k = 0 \quad \forall k \in K \\ p'_l = p_l - \frac{\epsilon}{2b(N-|K|)} \quad \forall l \in L \\ \mu'_j = \mu_j = 0 \quad \forall j \in M \\ \mu'_k = \mu_k + \frac{\epsilon}{N-|K|} \quad \forall k \in K \\ \mu'_l = \mu_l = 0 \quad \forall l \in L \end{cases}$$

if $\epsilon < 0$.

From the above, we can see that the prices of all outcomes are continuous while we changing q_i in its ϵ -neighborhood. However, $\partial p_i/\partial q_i$, $\partial p_j/\partial q_i$, and $\partial p_l/\partial q_i$ are not defined at \vec{q} for all $i, j \in M$, $i \neq j$, and $l \in L$, because the left and right derivatives do not equal. We only have $\partial p_k/\partial q_i = 0$ for all $k \in K$ and $i \in M$.

- $L \neq \emptyset$, $l \in L$

When changing \vec{q} to \vec{q}' , where $q'_l = q_l + \epsilon$ and $q'_j = q_j \forall j \neq l$, similar to the above case, the new optimal solution is different for $\epsilon > 0$ and $\epsilon < 0$. In particular, when $\epsilon > 0$,

$$p'_l = p_l + (\frac{1}{2b} - \frac{1}{2b(N-|K|)})\epsilon,$$

$$p'_j = p_j - \frac{\epsilon}{2b(N-|K|)} \quad \forall j \in M \cup L \text{ such that } j \neq l,$$

and $p'_k = p_k = 0 \forall k \in K$. When $\epsilon < 0$, $p'_j = p_j \forall j$, $\mu'_l = \mu_l - \epsilon$, and $\mu'_j = \mu_j \forall j \neq k$.

We can see that the prices of all outcomes are continuous while we changing q_l in its ϵ -neighborhood. But $\partial p_l/\partial q_l$ and $\partial p_j/\partial q_l$ are not defined at \vec{q} for all $j \in M \cup L$, $j \neq l$, and $l \in L$, because the left and right derivatives do not equal. $\partial p_k/\partial q_l = 0$ for all $k \in K$ and $l \in L$.

- $k \in K$

Consider changing \vec{q} to \vec{q}' , where $q'_k = q_k + \epsilon$ and $q'_j = q_j \forall j \neq k$, the new prices are defined by $p'_j = p_j \forall j$, $\mu'_k = \mu_k - \epsilon$, and $\mu'_j = \mu_j \forall j \neq k$. Thus, we have $\partial p_k/\partial q_k = 0 \forall k \in K$, $\partial p_j/\partial q_k = 0 \forall k, j \in K$, and $\partial p_j/\partial q_k = 0 \forall j \in M, k \in K$.

The above shows that the price functions are continuous everywhere, but not differentiable everywhere. In particular, when L is not empty (i.e., given \vec{q} there exists some outcome i such that both $p_i(\vec{q})$ and $\mu_i(\vec{q})$ are zero), some of the partial derivatives are not defined at \vec{q} . This corresponds to the second and third cases shown above. We further note that in these two cases, for any $i \in M$ and $l \in L$, any change in q_i or q_l will cause the set of L to become empty, because either p_l or μ_l will become positive for all $l \in L$. This means that the prices are differentiable almost everywhere, with the only exceptions at finite number of points when $L \neq \emptyset$.

Because $0 \leq |K| \leq N-1$, $0 \leq D_{i,i}(t) \leq 1/(2b) - 1/(2bN)$ and $-1/(2b) \leq D_{i,j}(t) \leq 0$. We have

$$\sum_{i=1}^N \sum_{j=1}^N |D_{i,j}(t)| \leq N \left(\frac{1}{2b} - \frac{1}{2bN} \right) + N(N-1) \frac{1}{2b} = \frac{N^2 - 1}{2b}.$$

□