An Optimization-Based Framework for Combinatorial Prediction Market Design[*]

Jacob Abernethy
UC Berkeley
jake@cs.berkeley.edu

Yiling Chen
Harvard University
yiling@eecs.harvard.edu

Jennifer Wortman Vaughan
UCLA
jenn@cs.ucla.edu

Abstract

We build on ideas from convex optimization to create a general framework for the design of efficient prediction markets over very large outcome spaces.

1 Introduction

A prediction market is a financial market primarily focused on the aggregation of information. To facilitate trades, prediction markets are often operated by automated market institutions. The market institution, referred to as a *market maker*, trades a set of securities. In the most simple case, each security corresponds to the outcome of an event. The market maker might offer a security that pays off $\$1$ if and only if BP files for bankruptcy by the end of the year. A risk neutral trader who believes that the probability of BP filing for bankruptcy is p should be willing to purchase this security at any price below p, or sell it at any price above p. Based on this intuition, the current market price can be viewed as the traders' collective estimate of how likely it is that BP will file for bankruptcy. Market-based probability estimates have proved to be accurate in a variety of domains including business, entertainment, and politics [2, 13, 17].

Prediction market research has largely focused on *cost function based markets* over *complete* security spaces [5, 6]. Consider a future event with n mutually exclusive and exhaustive potential outcomes, such as a contest with n possible winners. In a complete cost function based market, a market maker buys and sells securities corresponding to each outcome $i \in \{1, \cdots, n\}$. The security associated with outcome i pays out $1 if i is the final outcome, and $0 otherwise. The market maker determines how much each security should cost using a differentiable *cost function*. A cost function $C : \mathbb{R}^n \to \mathbb{R}$ is simply a potential function specifying the amount of money currently wagered in the market as a function of the number of outstanding securities. If q_i is the number of securities on outcome i currently held by traders, and a trader would like to purchase a bundle of r_i securities for each i (where some r_i could be 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 i (that is, the price per security of an infinitely small portion of a security) is then $\partial C(\mathbf{q})/\partial q_i$, and is denoted $p_i(\mathbf{q})$.

The market designer is free to choose any differentiable cost function C that satisfies a few basic properties. First, it must be the case that for every $i \in \{1, \cdots, n\}$ and every $\mathbf{q} \in \mathbb{R}^n$, $p_i(\mathbf{q}) \geq 0$. This ensures that the price of a security is never negative. Second, it must be the case that for every $\mathbf{q} \in \mathbb{R}^n$, $\sum_{i=1}^{n} p_i(\mathbf{q}) = 1$. If the instantaneous 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. These conditions ensure that there are no arbitrage opportunities within the market. They also ensure that the current prices can be viewed as a probability distribution over the outcome space, representing the market's prediction based on the traders' collective beliefs.

When the number of outcomes is very large, it might not be feasible to run a complete prediction market over the full outcome space. There has been a surge of recent research examining the

[*] For more details, see our longer draft available at http://arxiv.org/abs/1011.1941.

tractability of running standard prediction market mechanisms (such as the popular Logarithmic Market Scoring Rule [10]) over combinatorial outcome spaces by limiting the space of available securities [7–9, 15]. For example, if the outcome space contains all possible $n!$ rank orderings of n horses in a race, a market maker might choose to sell only pairwise securities of the form "horse A will finish the race ahead of horse B." If the outcome space is a large Boolean space, the market maker might sell securities on disjunctions of two events ("either a Democrat will win the 2012 senate race in Delaware or a Democrat will win in Ohio"). In these examples, running a naive implementation of the Logarithmic Market Scoring Rule over the full outcome space would be infeasible.

While this line of research has led to a few positive results (most notably, the tractability of pricing certain types of securities for large, single-elimination tournaments [8]), it has led more often to hardness results [7] or to markets with undesirable properties such as unbounded worst case market maker loss [9]. Building on recent work exploring mathematical connections between prediction market mechanisms and algorithms for online learning [6], we take a different approach to market design. Instead of beginning with an existing market mechanism and searching for a security space for which it is tractable to calculate prices, we incorporate ideas from online convex optimization [11, 16] to design *new* cost function based markets tailored to any security space we have in mind. This approach is more general and applies to a wide variety of settings.

2 A New Framework for Market-Making Over Complex Security Spaces

In the complete cost function based markets described above, the market maker offers a security corresponding to each potential state of the world. We consider a market-design scenario where the state space $\mathcal{O}$ could potentially be quite large, or even infinite, making it infeasible to run such a market. Instead, we allow the market maker to offer a menu of K securities for some reasonably-sized K, with the payoff of each security described by an arbitrary but efficiently-computable function $\boldsymbol{\rho} : \mathcal{O} \to \mathbb{R}_+^K$. Specifically, if a trader purchases a share of security i and the outcome is $\mathfrak{o}$, then the trader is paid $\rho_i(\mathfrak{o})$. We call such security spaces *complex*. A complex security space reduces to the complete security space if $K = |\mathcal{O}|$ and for each $i \in \{1, \cdots, K\}$, $\rho_i(\mathfrak{o}) = 1$ if and only if $\mathfrak{o}$ is the ith outcome. We consider traders that purchase *security bundles* $\mathbf{r} \in \mathbb{R}^K$. The payoff for $\mathbf{r}$ upon outcome $\mathfrak{o}$ is exactly $\boldsymbol{\rho}(\mathfrak{o}) \cdot \mathbf{r}$, where $\boldsymbol{\rho}(\mathfrak{o})$ denotes the vector of payoffs for each security for outcome $\mathfrak{o}$. Let $\boldsymbol{\rho}(\mathcal{O}) = \{\boldsymbol{\rho}(\mathfrak{o}) | \mathfrak{o} \in \mathcal{O}\}$.

We do not presuppose a cost function based market. However, we show that the use of a convex potential function is *necessary* given some minor assumptions.

2.1 Imposing Some Natural Restrictions on the Market Maker

In this section we introduce a sequence 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 the long version.) Similar conditions were suggested for complete markets by Chen and Vaughan [6], who defined the notion of a *valid cost function*, and by Othman et al. [14], who discussed properties similar to our notions of path independence and expressiveness, among others.

Imagine a sequence of traders entering the marketplace and purchasing security bundles. Let $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \ldots$ be the sequence of security bundles purchased. After $t-1$ such purchases, the tth 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 $\texttt{Cost}(\mathbf{r}|\mathbf{r}_1, \ldots, \mathbf{r}_{t-1})$ for any bundle $\mathbf{r}$. If the trader chooses to purchase $\mathbf{r}_t$ at a cost of $\texttt{Cost}(\mathbf{r}_t|\mathbf{r}_1, \ldots, \mathbf{r}_{t-1})$, 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 $\mathbf{r}_1, \ldots, \mathbf{r}_t$,* $\texttt{Cost}(\mathbf{r}|\mathbf{r}_1, \ldots, \mathbf{r}_t) = \texttt{Cost}(\mathbf{r}'|\mathbf{r}_1, \ldots, \mathbf{r}_t) + \texttt{Cost}(\mathbf{r}''|\mathbf{r}_1, \ldots, \mathbf{r}_t, \mathbf{r}')$.

It turns out that this condition *alone* implies that prices can be represented by a cost function C, as illustrated in the following theorem. The proof is by induction on t.

Theorem 1. *Under Condition 1, there exists a cost function $C : \mathbb{R}^K \to \mathbb{R}$ such that we may always write* $\texttt{Cost}(\mathbf{r}_t|\mathbf{r}_1, \ldots, \mathbf{r}_{t-1}) = C(\mathbf{r}_1 + \ldots + \mathbf{r}_{t-1} + \mathbf{r}_t) - C(\mathbf{r}_1 + \ldots + \mathbf{r}_{t-1})$.

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

We would like to aggregate traders' beliefs into an accurate prediction. Each trader may have his own (potentially secret) information about the future, which we can represent as some distribution $\mathbf{p} \in \Delta_{|\mathcal{O}|}$ over the outcome space. The pricing mechanism should therefore incentivize the traders to reveal $\mathbf{p}$, while simultaneously avoid providing arbitrage opportunities. Towards this goal, we introduce 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 infinitesimally small quantity of any 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 infinitesimally small 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 $\partial C(\mathbf{q})/\partial q_i$ was interpreted as the probability of the ith outcome for the complete security space.

Condition 2 (Existence of Instantaneous Prices)**.** *C is continuous and differentiable everywhere.*

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 $\mathbf{r}$. 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 $-\mathbf{r}$.

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.

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

Finally, expressiveness specifies that a trader can set the market prices to reflect his beliefs about the expected payoffs of each security if arbitrarily small portions of shares may be purchased.

Condition 5 (Expressiveness)**.** *For any $\mathbf{p} \in \Delta_{|\mathcal{O}|}$, $\exists \mathbf{q} \in \mathbb{R}^K \cup \{\infty, -\infty\}$ for which $\nabla C(\mathbf{q}) = \mathbb{E}_{\mathfrak{o}\sim\mathbf{p}}[\boldsymbol{\rho}(o)]$.*

Let $\mathcal{H}(S)$ denote the convex hull of a set $S \subset \mathbb{R}^K$. We characterize the form of the cost function under these conditions.

Theorem 2. *Under Conditions 2-5, C must be convex with $\{\nabla C(\mathbf{q}) : \mathbf{q} \in \mathbb{R}^K\} = \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$.*

Specifically, the existence of instantaneous prices implies that $\nabla C(\mathbf{q})$ is well-defined. The incorporation of information condition implies that C is convex. The convexity of C and the no arbitrage condition imply that $\{\nabla C(\mathbf{q}) : \mathbf{q} \in \mathbb{R}^K\} \subseteq \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$. Finally, the expressiveness condition is equivalent to requiring that $\mathcal{H}(\boldsymbol{\rho}(\mathcal{O})) \subseteq \{\nabla C(\mathbf{q}) : \mathbf{q} \in \mathbb{R}^K\}$.

This theorem tells us that to satisfy our conditions, the set of reachable prices of a market should be *exactly* the convex hull of $\boldsymbol{\rho}(\mathcal{O})$. For complete markets, this would imply that the set of reachable prices should be precisely the set of all probability distributions over the n outcomes.

2.2 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 $\boldsymbol{\rho} : \mathcal{O} \to \mathbb{R}_+^K$, we should use a cost function based market with a convex, differentiable cost function such that $\{\nabla C(\mathbf{q}) : \mathbf{q} \in \mathbb{R}^K\} = \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$. We now provide a general technique that can be used to design and compare properties of cost functions that satisfy these criteria, using tools from convex analysis.

It is well known that any closed, convex, differentiable function $C : \mathbb{R}^K \to \mathbb{R}$ can be written in the form $C(\mathbf{q}) = \sup_{\mathbf{x}\in\mathrm{dom}(R)} \mathbf{x} \cdot \mathbf{q} - R(\mathbf{x})$ for a strictly convex function R called the *conjugate* of C [3, 12]. (The strict convexity of R follows from the differentiability of C.) Furthermore, any function that can be written in this form is convex. As we describe in the full version of this work, the gradient of C can be expressed in terms of this conjugate: $\nabla C(\mathbf{q}) = \operatorname{argmax}_{\mathbf{x}\in\mathrm{dom}(R)} \mathbf{x}\cdot\mathbf{q} - R(\mathbf{x})$. To generate a convex cost function C such that $\nabla C(\mathbf{q}) \in \Pi$ for all $\mathbf{q}$ for some set Π, it is therefore sufficient to choose a conjugate function R, restrict the domain of R to Π, and define C as

$$C(\mathbf{q}) = \sup_{\mathbf{x}\in\Pi} \mathbf{x}\cdot\mathbf{q} - R(\mathbf{x})\ . \tag{1}$$

We call such a market a *complex cost function based market*. To generate a cost function C satisfying our five conditions, we need only to set $\Pi = \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$ and select a strictly convex function R.

This method of defining C is convenient for several reasons. First, it leads to markets that are efficient to implement whenever Π can be described by a polynomial number of simple constraints. Similar techniques have been applied to design learning algorithms in the online convex optimization framework [11, 16], where R plays the role of a regularizer, and have been shown to be efficient in a variety of combinatorial applications, including online shortest paths, online learning of perfect matchings, and online cut set [4]. Second, it yields simple formulas for properties of markets that help us choose the best market to run, such as worst-case monetary loss and information loss.

Note that both the LMSR and Quad-SCPM [1] are examples of complex cost function based markets, though they are designed for the complete market setting only.

2.3 An Example

To illustrate the use of our framework for market design, we consider the following example. 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 $\mathbf{u} \in \mathbb{R}^3$. 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 $\mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$ for this market will be the ℓ_2-norm unit ball centered at $\mathbf{1}$. We shall use $\|\cdot\|$ to refer to the ℓ_2-norm. 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 $\|\mathbf{q}\| \le 2\lambda$, 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 $\mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$. Specifically, it is minimized at $\mathbf{x} = \mathbf{q}/||\mathbf{q}|| + \mathbf{1}$. From this, we can compute

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

We can show that the worst-case monetary loss of the market maker is no more than 2λ, and the information loss, defined as the bid-ask spread of a bundle $\mathbf{r}$, scales linearly with $\|\mathbf{r}\|^2/\lambda$; see the longer version for details.

By relaxing our “no-arbitrage” condition, we can also use this framework to design a new efficient market maker for pair betting (“horse A ahead of horse B”), which is known to be #P-hard to price using LMSR [7]. Surprisingly, this relaxation does not increase the market maker’s worst-case loss, and can actually lead to a profit.

3 Conclusion

Leveraging techniques from convex optimization, we propose a general framework to design market maker mechanisms on arbitrary security spaces. While past research on combinatorial prediction markets has focused on finding security spaces that are tractable to price using popular market mechanisms, our framework opens up the possibility of designing *new* efficient market maker mechanisms for security spaces of interest, such as pair betting markets. We believe that this framework will lead to fruitful new directions of research in prediction market design and implementation.

References

[1] S. Agrawal, E. Delage, M. Peters, Z. Wang, and Y. Ye. A unified framework for dynamic prediction market design. *Operations Research*, to appear, 2010.

[2] 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.

[3] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[4] N. Cesa-Bianchi and G. Lugosi. Combinatorial bandits. In submission, January 2010.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] R. Hanson. Combinatorial information market design. *Information Systems Frontiers*, 5(1): 105–119, 2003.

[11] E. Hazan. A survey: The convex optimization approach to regret minimization. Draft, Sept. 2009.

[12] J.-B. Hiriart-Urruty and C. Lemaréchal. *Fundamentals of Convex Analysis*. Springer, 2001.

[13] J. Ledyard, R. Hanson, and T. Ishikida. An experimental test of combinatorial information markets. *Journal of Economic Behavior and Organization*, 69:182–189, 2009.

[14] 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.

[15] 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.

[16] A. Rakhlin. Lecture notes on online learning. Draft, April 2009.

[17] J. Wolfers and E. Zitzewitz. Prediction markets. *Journal of Economic Perspective*, 18(2): 107–126, 2004.