Connections Between Markets and Learning

YILING CHEN
Harvard University
and
JENNIFER WORTMAN VAUGHAN
Harvard University & UCLA

We provide an overview of recent research exploring the striking mathematical connections that exist between market maker mechanisms for prediction markets and no-regret learning. We describe how these connections can be used in the design of efficient prediction markets over combinatorial outcome spaces.

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

General Terms: Algorithms, Economics, Theory

1. INTRODUCTION

A *prediction market* is a financial market designed to aggregate information. In a typical prediction market, the organizer (or *market maker*) trades a set of securities, each corresponding to a potential outcome of an event. The market maker might offer a security that will pay off \$1 if and only if Sarah Palin becomes the U.S. Republican Party presidential nominee in 2012. A risk neutral trader who believes that the probability of Palin winning the nomination is p should be willing to buy a share of this security at any price below \$$p$. Similarly, he should be willing to sell at any price above \$$p$. For this reason, the current market price can be viewed as the population's collective estimate of how likely it is that Palin will be the nominee.

These market-based estimates have proved to be accurate in a variety of domains. Election markets can be more accurate than polls, Oscar markets more accurate than expert columnists, and internal corporate markets more accurate than professional sales forecasts; see Ledyard et al. [2009] for an impressive assortment of additional examples. The theory of rational expectations equilibria offers an explanation of why prediction markets should yield accurate forecasts at equilibrium (at least under strong assumptions about the traders' beliefs and behavior), but says nothing about why particular market mechanisms, such as Hanson's popular Logarithmic Market Scoring Rule [2003], might produce more accurate estimates than others. This letter describes a first step toward developing stronger insight into the learning power of particular market mechanisms by examining the deep mathematical connections between prediction markets and no-regret learning. We begin with a review of cost function based markets and the well-studied problem of "learning from expert advice" before exploring these connections in more detail.

Authors' addresses: yiling@eecs.harvard.edu, jenn@seas.harvard.edu. Based on a longer paper that will appear at the 11th ACM Conference on Electronic Commerce [Chen and Vaughan 2010].

2. COST FUNCTION BASED MARKETS

Consider a set of N mutually exclusive and exhaustive outcomes of a future event, such as N competing horses winning a race. In a cost function based market, an automated market maker offers a security corresponding to each outcome $i \in \{1, \cdots, N\}$. The security associated with outcome i is guaranteed to pay off \$1 if i happens, and \$0 otherwise. Different mechanisms can be used to determine how these securities are priced, each mechanism specified using a differentiable *cost function* $C : \mathbb{R}^N \rightarrow \mathbb{R}$. This cost function can be viewed as 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 security i (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 infinitesimally small number of shares) is then $p_i = \partial C(\vec{q})/\partial q_i$.

A few restrictions must be made on the cost function C to ensure that the resulting market makes sense. In particular, we say that a cost function is *valid* if it satisfies three simple properties: differentiability (to ensure that the instantaneous prices are well-defined), increasing monotonicity (to ensure that the instantaneous prices are never negative), and a simple “translation invariance” property that guarantees that the cost of purchasing one share of all N outcomes is always \$1 (to protect against arbitrage). It turns out that for any valid cost function, the set of instantaneous prices will always form a probability distribution over the outcome space. As described above, the price of outcome i can then be viewed as representing the market’s current estimate of the probability that i will occur.

Hanson introduced the class of *market scoring rule* (MSR) markets [2003], in which traders sequentially change the market’s probability estimates and receive payments according to their incremental improvements to the market’s prediction evaluated by a proper scoring rule. Any market in this class can be implemented as a cost function based market; we refer the interested reader to Chen and Vaughan [2010] for the exact translation. The popular *logarithmic market scoring rule* (LMSR) has a cost function $C(\vec{q}) = b \log \sum_{i=1}^{N} \mathrm{e}^{q_i/b}$ and prices proportional to $\mathrm{e}^{q_i/b}$, where b is a positive free parameter controlling the market liquidity [Chen and Pennock 2007]. The LMSR satisfies the nice property that the worst-case loss of the market maker is bounded by $b \log N$. Bounded market maker loss is clearly a desirable property in practice. All MSR markets using *regular* proper scoring rules (that is, scoring rules that are finite except when the reported probability of an outcome is zero, in which case the score for the outcome can be negative infinity), and hence the corresponding cost function based markets, have bounded market maker loss.

3. LEARNING FROM EXPERT ADVICE

No-regret learning has been studied with great interest in many communities under many different names since as early as the 1950s. We consider the basic “learning from expert advice” framework, in which an algorithm makes a sequence of predictions based on the advice of a set of N *experts* (which can be individual features, weak learners, human advisers, or any other forecasters) and receives a corresponding sequence of *losses*. The goal of the 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, the algorithm must perform well even if the sequence of losses is chosen by an adversary.

Formally, at every time step $t \in \{1, \cdots, T\}$, every expert $i \in \{1, \cdots, N\}$ 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, $L_{\mathcal{A},T} - \min_{i \in \{1,\cdots,N\}} L_{i,T}$. 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 [Littlestone and Warmuth 1994; Freund and Schapire 1997] is an example of a no-regret algorithm. Weighted Majority uses weights $w_{i,t}$ proportional to $\mathrm{e}^{-\eta L_{i,t}}$, 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 by $\eta T + \log(N)/\eta$. When T is known in advance, setting $\eta = \sqrt{\log N/T}$ yields the standard no-regret bound of $O(\sqrt{T \log N})$.

4. MAKING THE CONNECTION

A careful reader may have noticed that the weights employed by Weighted Majority look very similar to the LMSR price function described in Section 2. Chen et al. [2008] were the first to draw a formal connection between WM and LMSR. In particular, they showed that the WM regret bound can be used as a starting point to rederive the bound of $b \log N$ on the worst-case loss of the LMSR market maker. In our recent work, we showed that the converse holds as well: the WM regret bound can be derived as a consequence of the LMSR market maker’s bounded loss. However, the connection goes much deeper.

In fact, *any* cost function based prediction market with bounded loss can be interpreted as a no-regret learning algorithm. Furthermore, the bound on the market maker’s loss can always 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. (Note that it is *not* the traders in the market that are analogous to the experts, as one might initially guess!) 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.

More specifically, consider the following. A learning algorithm maintains weights

$w_{i,t}$ for each of a set of N experts that depend on the total loss $L_{i,t}$ of each expert i. The algorithm's regret is measured as the difference between its own cumulative loss and the cumulative loss of the best expert, $\min_{i \in \{1,\cdots,N\}} L_{i,t}$. Analogously, the market maker maintains prices $p_{i,t}$ for each of a set of N outcomes that depend on the total quantity of shares purchased for each outcome i (call this $Q_{i,t}$). We are interested in measuring the market maker's maximum loss, which is the difference between the quantity of money he has collected and the maximum number of shares purchased on any outcome, $\max_{i \in \{1,\cdots,N\}} Q_{i,t}$.

Suppose we have a particular cost function based market in mind. We can build a no-regret learning algorithm by simulating this market, using the expert losses as quantities of shares sold. It is straight-forward to relate the loss of the best expert with the maximum number of shares purchased on any outcome in the simulated market. To relate the learning algorithm's regret to the market maker's worst-case loss, it is therefore only necessary to relate the loss of the learning algorithm to the amount of money collected by the market. Luckily, if the market prices change sufficiently slowly (i.e., if the market liquidity is sufficiently high), these two quantities will be similar. Using these ideas, we can show that the regret of the resulting learning algorithm is bounded by $O(\sqrt{B\phi T})$, where B is the worst-case market maker loss and ϕ is a measure of how fast the market prices change.

5. DIGGING DEEPER

It may seem surprising that *any* cost function based market with bounded loss (and thus any market scoring rule) can be used to derive a learning algorithm that satisfies the nice property of no regret. It is natural to wonder what the resulting algorithms look like. It turns out that the set of expert learning algorithms derived from valid convex cost functions share a common form. In particular, they are precisely a recently discovered class of algorithms known as "Follow the Regularized Leader" algorithms. (See, for example, Hazan and Kale [2008].) This class of algorithms grew out of the following fundamental insight.

Consider first the aptly named *Follow the Leader* algorithm, which chooses weights at time t to minimize the empirical loss $\sum_{i=1}^{N} w_{i,t} L_{i,t-1}$. 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 construct examples in which the best expert changes frequently, causing Follow the Leader to suffer $\Omega(T)$ regret.

To overcome this instability, one can add a perturbation to the empirical loss of each expert, and choose the expert that minimizes this perturbed loss. It is possible to gain the necessary stability by adding what is called a *regularizer* $\mathcal{R}$ and choosing weights to minimize

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

In general, choosing weights in this manner will prevent all of the algorithm's weight from being placed on a single expert. WM can be written in this form with an entropy term as the regularizer.

This Follow the Regularized Leader (FTRL) approach gets around the instability of FTL and guarantees low regret for a wide variety of regularizers. In our analogy between markets and learning, the choice of a cost function in the market corresponds to the choice of a regularizer in the learning problem.

6. APPLYING THE CONNECTION

In this letter, we have hinted at the elegant mathematical connection between prediction markets and no-regret learning. While this connection is thought-provoking on its own, we believe that its true power lies in its 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 [Pennock and Sami 2007; Chen et al. 2008; Chen et al. 2008]. 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 second 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. By exploiting the connection between Weighted Majority and the LMSR, Chen et al. [2008] showed that an extension of Weighted Majority could be used to approximate prices in an LMSR over permutations, where computing prices exactly is #P-hard. 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, we are hopeful that no-regret learning will provide the tools needed to design specialized markets for many other scenarios in which the outcome space is large.

REFERENCES

Chen, Y., Fortnow, L., Lambert, N., Pennock, D. M., and Wortman, J. 2008. Complexity of combinatorial market makers. In *ACM EC*.

Chen, Y., Goel, S., and Pennock, D. M. 2008. Pricing combinatorial markets for tournaments. In *ACM STOC*.

Chen, Y. and Pennock, D. M. 2007. A utility framework for bounded-loss market makers. In *UAI*.

Chen, Y. and Vaughan, J. W. 2010. A new understanding of prediction markets via no-regret learning. In *ACM EC*.

Freund, Y. and Schapire, R. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Comp. and System Sciences 55,* 1, 119–139.

Hanson, R. 2003. Combinatorial information market design. *Information Systems Frontiers 5,* 1, 105–119.

Hazan, E. and Kale, S. 2008. Extracting certainty from uncertainty: Regret bounded by variation in costs. In *COLT*.

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

Littlestone, N. and Warmuth, M. 1994. The weighted majority algorithm. *Info. and Computation 108,* 2, 212–261.

Pennock, D. M. and Sami, R. 2007. Computational aspects of prediction markets. In *Algorithmic Game Theory*, N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, Eds. Cambridge University Press.