UPenn NETS 412: Algorithmic Game Theory	April 3, 2018
Lecture 15	
Lecturer: Bo Waggoner	*Scribe: Bo Waggoner*

Aggregating Information: Prediction Markets

Now we will go back to the problem of predicting a future event, i.e. a case where we can verify predictions after the fact. This time, however, we will have multiple agents, each with different pieces of information or beliefs. The goal will be to aggregate these into a single prediction.

One thing we can do is ask them all to simultaenously report a prediction, and score each one with a proper scoring rule. This does elicit each's belief truthfully, but there are some drawbacks:

- It may be expensive or wasteful: If the experts all agree, then we pay many times for the same prediction/information.
- On the other hand, if the experts give many different predictions, then it isn't clear how to **aggregate** them into a single prediction. Should we just average them? (Can you think of a scenario where this would be a bad idea?)
- In particular, it might require expert knowledge to correctly aggregate the predictions.

Robin Hanson had an excellent idea for addressing both of these in his paper *Combinatorial Information Market Design* (2003). Hanson called this a *market scoring rule* form of a "prediction market". There are also other forms of prediction markets, which we will see toward the end of these notes.

Scoring rule based markets. The idea is to approach the experts *sequentially* and ask them to make a prediction, given the predictions that previous experts have made. However, they are expected to "improve" on the prediction: they will get the score of their prediction *minus* the score of the previous one. More formally, suppose we wish to predict an event with outcomes $\mathcal{Y}$. Recall that $\Delta_{\mathcal{Y}}$ is shorthand for the set of probability distributions on $\mathcal{Y}$.

1. The designer chooses a proper scoring rule S and an initial prediction $p^0 \in \Delta_{\mathcal{Y}}$.
2. The experts $t = 1, \dots, T$ arrive in order. Each expert t changes the prediction from p^{t-1} to p^t.
3. The market ends and the event's outcome is observed, say, y.
4. Each expert t receives a payoff
$$S(p^t, y) - S(p^{t-1}, y).$$

This is a very simple idea, given that we already know about proper scoring rules, but it's also pretty brilliant. We will see that it has a number of nice properties, which lead us to believe that such a market can do a good job of aggregating information in practice (which does seem to be the case empirically).

Theorem 1 *If S is a proper scoring rule, and each agent arrives only once, then each agent maximizes expected score by reporting her true belief given her own knowledge and the trades of the previous experts.*

Proof The tth expert will be scored with $S(p^t, y) - S(p^{t-1}, y)$. The second term is a constant not in her control, so her score is $S(p^t, y) - \beta$ for some constant β, which is a proper scoring rule. So she maximizes expected score by reporting her true belief at the time she is asked. ■

Note part of the key idea here is that the expert should update her belief to include all the actions of the previous experts. If everyone else is predicting that the event will not happen, perhaps they hold important information and she should change her own beliefs.

(However, this result assumes crucially that each agent only participates **once**. If they can participate multiple times, then they may have an incentive to misreport early and only truthfully report later. There is ongoing research on when this might happen and how to prevent it.)

A main benefit provided by these scoring rule markets is that the total payment the designer has to make to all the agents is bounded.

Theorem 2 *For any proper scoring rule S, the designer can choose the initial prediction p^0 so that total payment is bounded by a constant independent of T (the number of arrivals).*

Proof The total payment is the sum over the agents of the payments made to each:

$$\begin{aligned} &S(p^1, y) - S(p^0, y) + S(p^2, y) - S(p^1, y) + \ldots + S(p^T, y) - S(p^{T-1}, y) \\ &= S(p^T, y) - S(p^0, y). \end{aligned}$$

This is the key point: the telescoping sum. This total payment does not depend on T; it only depends on the initial and final predictions as well as the outcome. So this proves the theorem.

We can get more detailed bounds on the total payment: $S(p^T, y) \leq S(\delta_y, y) = f(\delta_y)$ for some convex function f (we are assuming $f(\delta_y) \neq \infty$). So this gives an upper bound of $\max_y f(\delta_y) - S(p^0, y)$, which, again, does not depend on the number T of agents.

Going further, by choosing p^0 to be e.g. the uniform distribution, we could bound $S(p^0, y)$ by a constant only depending on the gradient of f at p^0. ■

These markets also have a final good incentive property: no arbitrage. *Arbitrage* is an opportunity to guarantee oneself to make money, i.e. in a scoring rule based prediction market, it is some update from p^{t-1} to p^t such that, for all y, the agent makes a strictly positive profit.

Theorem 3 *In a scoring rule market, there is never an opportunity for arbitrage: for all p^{t-1}, p^t, there exists an outcome y such that $S(p^t, y) - S(p^{t-1}, y) \leq 0$.*

Proof We will first show that there exists a probability distribution over $\mathcal{Y}$ such that the expected payoff for this trade is nonpositive. This will imply that there exists some outcome y where the agent loses money.

Suppose the outcome is drawn from distribution p^{t-1}, i.e. the previous agent was exactly correct about the distribution. Then expected payoff for this update is $S(p^t; p^{t-1}) - S(p^{t-1}; p^{t-1})$. (Remember the notation $S(p; q)$ refers to expected score for reporting p when $y \sim q$.)

But by properness of the scoring rule, if $y \sim p^{t-1}$, then the expected score is higher for reporting p^{t-1}, i.e. $S(p^{t-1}; p^{t-1}) \geq S(p^t; p^{t-1})$. So $S(p^t; p^{t-1}) - S(p^{t-1}; p^{t-1}) \leq 0$.

We have found a distribution for y where the agent's expected payoff is not positive. The expected payoff is just a weighted average over the scores for different outcomes y. More precisely,

$$S(p^t; p^{t-1}) - S(p^{t-1}; p^{t-1}) = \sum_{y \in \mathcal{Y}} p^{t-1}(y) \left(S(p^t, y) - S(p^{t-1}, y) \right).$$

Since this average is at most 0, there must be some term that is at most zero, i.e. there is some y such that $S(p^t, y) - S(p^{t-1}, y) \leq 0$. ■

Price-based markets and connections to online learning. Real-world prediction markets work more like stock markets: Agents arrive and can buy or sell "shares" of "securities", where each security is named after a possible outcome in $\mathcal{Y}$ and a share in the security for y pays off 1 if $Y = y$ and 0 otherwise.[1] The idea is that the price of the security should approach a "collective prediction" for the probability of the event, since the probability of the event is exactly the expected value of a share.

[1]There are also more complex securities with more complicated payoff functions, but that's beyond the scope of this lecture.

For a simple example, suppose we wish to predict whether Home Team wins the Championship. We can offer two securities, "Win" and "Lose". The initial prices of the securities is some $p \in \Delta_{\{\text{Win, Lose}\}}$ such as $(0.6, 0.4)$. If someone thinks their team is very likely to win, they can buy one share of the "Win" security for a price of 0.6. If the team wins, that share gives them a payoff of 1. They only paid 0.6, so this is a profit of 0.4.

Note that if their belief was a probability of 0.7, for example, then their expected payoff from one share is $0.7(1) + 0.3(0) = 0.7$. So they expect to make a profit (on average) from buying a share if and only if their belief is higher than the current price.

Also note that we could allow them to purchase fractions of a share. If they buy 0.3 shares and the event happens, they get 0.3 payoff. The pay 0.3 times the price.

More formally:

1. The market maker sets initial prices $p^0 \in \Delta_{\mathcal{Y}}$. That is, the current price for a share of outcome y is p^0_y, the probability assigned to y.

2. For $t = 1, \ldots, T$, agent t arrives, observes the current prices, and picks a *bundle* x^t where, for every y, $x^t_y \in [0, 1]$. Here x^t_y is a number of *shares* purchased in outcome y. The agent pays the current prices for all the shares. So total payment is $x^t \cdot p^{t-1} = \sum_y x^t_y p^{t-1}_y$.

3. After each agent t arrives and purchases a bundle, the market maker updates the prices to p^t.

4. After all agents arrive, the event's outcome y is observed, and each agent is paid 1 unit per share they purchased in y. So agent t gets paid x^t_y.

Now, the market maker's total loss equals the payoffs of the agents for the outcome y, minus all the prices paid by the agents:

$$\begin{aligned}
&\sum_t x^t_y - \sum_t x^t \cdot p^{t-1} \\
&\leq \left(\max_y \sum_t x^t_y \right) \quad - \sum_t x^t \cdot p^{t-1} \\
&= \text{Regret}
\end{aligned}$$

So it turns out we can interpret this problem as picking a sequence of distributions $p^1, \ldots, p^T$ in order to minimize regret!

Note there are some slight differences from the definition of regret we saw in traditional online learning. There, each x^t was a vector of *losses* and regret was (loss of algorithm) $-$ (loss of optimal "action" y). Here, each x^t is a vector of *gains* and regret corresponds to (gain of opt) $-$ (gain of algorithm). The optimal prices in hindsight would have been δ_y every day, i.e. charge 1 for the otucome y that would end up happening and 0 for everything else.

We immediately get:

Theorem 4 *In the price-based market, by using the polynomial weights algorithm, the market maker can guarantee a loss of at most* $O(\sqrt{T})$.

And indeed, online learning is closely related to setting prices in real markets.

Connecting the two market approaches. We will end by stating, but not proving, a very interesting connection. Suppose, in the price-based market, the market maker was allowed to limit the trader to breaking up her purchase into smaller bundles of only $[0, \alpha]$ shares. After each purchase of up to α shares, the market maker gets to update her prices, then the trader can continue buying. In this case we have a surprising result:

Theorem 5 *As $\alpha \to 0$, the price-based market and scoring rule based markets become equivalent, in particular, the* log *scoring rule is equivalent to the polynomial weights algorithm.*

A brief sketch of the idea as follows:

- As $\alpha \to 0$, the polynomial weights algorithm converges to *exponential weights* where p_y^t is proportional to $e^{\sum_{s=1}^t x_y^s}$.
- Also, as $\alpha \to 0$, the agent's utility becomes $x_y^t - P$ where P is the total payment, which looks like the *integral* of the price as it changes from p^{t-1} to p^t.
- The log scoring rule would assign the agent a utility of $\log p_y^t - \log p_y^{t-1}$.
- This turns out to equal $x_y^t - P$ where P is a payment which equals the integral of prices from p^{t-1} to p^t. So the utilities are equivalent.

For details, the main paper to read is *Efficient Market Making via Convex Optimization, and a Connection to Online Learning* (2013) by Abernethy, Chen, and Wortman-Vaughan. The paper describes this equivalence in more generality.