

Summary about Prediction Market

Zhihao Ruan, Jiayu Yi, Naihao Deng
`{ruanzh,yijiaiyu,dnaohao}@umich.edu`

August 5, 2019

Contents

1	Exponential Family	2
2	Scoring Rules	3
2.1	Motivations	3
2.2	Incentive Compatibility	3
2.3	Logarithmic Scoring Rule	3
3	Cost Function based Prediction Market with Bayesian Traders	4
3.1	Bayesian Agents	4
3.2	Market Maker	4
3.3	Executable program	4

1 Exponential Family

2 Scoring Rules

Scoring rules is the simplest form of prediction mechanism. For every agent with some information, a scoring rule evaluates how close the information is from the actual outcome, and pays the agent in return for his/her information.

2.1 Motivations

Taking an event \mathbf{X} with outcome space \mathcal{X} , we want to know something about each agent's belief $p(\mathbf{x})$ on the actual outcome $\mathbf{x} \in \mathcal{X}$, compare it with the actual outcome, and see how accurate the agent's prediction is. However, it is impossible to ask agent for his/her entire $p(\mathbf{x})$ probability distribution. What should we do?

We've already known that sufficient statistic is a very suitable parameter to characterize a probability distribution. Therefore, we can use it to represent agent's belief. We just ask each agent for the sufficient statistic $\mathbf{u}(\mathbf{x})$ as a representation of his/her belief $p(\mathbf{x})$. Then we are able to measure how accurate the agent can predict with a scoring rule based on the actual outcome.

Assume $\hat{\boldsymbol{\mu}} = \mathbb{E}_p[\mathbf{u}(\mathbf{x})]$ is the *expected* sufficient statistic from agent's belief $p(\mathbf{x})$. Then, with the actual outcome denoted \mathbf{x} , the scoring rule takes the form:

$$S(\hat{\boldsymbol{\mu}}, \mathbf{x}).$$

We can see that the scoring rule is a measure of how close the agent's belief is from the actual outcome.

2.2 Incentive Compatibility

Incentive compatibility is a property of prediction mechanism with which the best strategy for an agent to earn the most profit is to *honestly* report all the information as soon as he/she has it. As a prediction mechanism, a proper scoring rule should leverage **incentive compatibility** in order to get real information from agents.

Assume that we already set the sufficient statistic to be $\mathbf{u}(\mathbf{x})$. Then for each $p \in \mathcal{P}$ that takes this $\mathbf{u}(\mathbf{x})$ as its sufficient statistic, we can calculate its *expected* sufficient statistic $\boldsymbol{\mu} = \mathbb{E}_p[\mathbf{u}(\mathbf{x})]$. A scoring rule is thus called **proper** if it satisfies the following, for all such p , for all $\hat{\boldsymbol{\mu}} \neq \boldsymbol{\mu}$:

$$\mathbb{E}_p[S(\boldsymbol{\mu}, \mathbf{x})] \geq \mathbb{E}_p[S(\hat{\boldsymbol{\mu}}, \mathbf{x})].$$

Any scoring rule with this property actually encourages agents to report a probability distribution as close to the actual probability distribution of \mathbf{X} as possible, which is in accordance with the essence of incentive compatibility.

2.3 Logarithmic Scoring Rule

Suppose that we have set a form of the sufficient statistic. A classic logarithmic scoring rule takes the form:

$$S(\boldsymbol{\mu}, \mathbf{x}) = \ln p(\mathbf{x}; \boldsymbol{\mu}),$$

where $\boldsymbol{\mu} = \mathbb{E}_p[\mathbf{u}(\mathbf{x})]$ is the expected sufficient statistic over $p(\mathbf{x}; \boldsymbol{\mu})$.

3 Cost Function based Prediction Market with Bayesian Traders

Based on what we have in previous sections, we could simulate the prediction market by python program.

For the setup, there would be a group of Bayesian agents (the number of agents as hyper-parameter would be an input for the program), a market maker and an executable program driving the market. They are explained in Sec 3.1, Sec 3.2 and Sec 3.3. The dataset the market is based on is subject to Bernoulli distribution.

3.1 Bayesian Agents

Each Bayesian agents would be initialized with several data points, based on which he would form his prior $p(\theta; b_0)$. The dataset is subject to Bernoulli distribution, thus we have $b_0 = \begin{bmatrix} n\nu \\ n \end{bmatrix}$, in which n would be the number of data points Bayesian agents have access to initially, $n\nu$ would be the sum of these data points.

Once the agent intends to enter the market, he would have access to several more data points before his entrance. He would update his belief by $b_1 = \begin{bmatrix} n\nu + m\hat{\mu} \\ n + m \end{bmatrix}$, that is, he would update the first entry of the vector as the sum of data he is now having access to, the second entry as the number of these data. He would also derive the amount of outstanding shares θ from the current market price, in our case, $\theta = -1/p$ (p is the current market price).

Based on

$$\nabla C(\theta + \delta) = \frac{n\nu + m\hat{\mu}}{n + m}.$$

, the agent would be able to decide δ , the amount he would like to trade by

$$\delta = -\frac{1}{\frac{n\nu + m\hat{\mu}}{n + m}} - \theta$$

.

3.2 Market Maker

The market maker is initialized with the initial outstanding security amount θ and the initial market price, which are hyper-parameter predefined.

The market maker holds the sufficient statistic $\phi(x) = x$, cost function for the market $C(\theta) = -\log(-\theta)$ and price function $p = -1/\theta$. He would keep track of number of trades, outstanding security amount and current market price when the market runs.

3.3 Executable program

The driving program takes agent number and max iteration number as hyper-parameters. The complete dataset is divided into two parts as random chosen data in the first part would be assigned to agents to form their priors, data in the second part would be fed to agents when they intend to enter the market.

Each agent comes into the market iteratively with the total number of iteration not exceeding max iteration number in program arguments. The interval between each agent's arrival is subject to Poisson distribution. As discussed in Sec 3.1 and Sec 3.2, each agent would update their belief and then trade with market maker. The final market price is considered as the aggregated belief among agents.