CS711: Introduction to Game Theory and Mechanism Design **Dec 5, 2020**

Project Report: Simulation of Prediction Market for IPL

Group 10 members: Akash Halayyanavar, Jay Kothadiya, Moksha Vora, Sagar Sharma, Tanisha Rastogi

Abstract

The objective of the project is to simulate the real-world like situation of the prediction market by simulating the investors and the automated market maker. The event considered for prediction in this project is: "Will Mumbai Indians (MI) win the IPL 2020?". The automated market maker is an entity regulating the prediction market by performing the buying and selling of the assets. The automated market maker in this project uses logarithmic market scoring rule to elicit the true information from the investors. The investors are simulated using machine learning model. The investors are considered to be of three types: Expert, Medium-knowledged and Novice, to replicate the real market scenario. The investor's belief regarding an event is converted into number of shares (to be bought or sold) by considering the current market scenario. The final outcome of the event is also known at this moment, so the step of market resolution is also performed to exactly implement the prediction market. Different observations are gathered by running this simulation under different scenarios (by changing the proportion of three types of investors in the market).

1 Introduction to the problem

Prediction markets are exchange-traded markets created for the purpose of trading the outcome of events. They can be thought of as belonging to the more general concept of crowdsourcing. The main objective of the prediction market is to predict the probability of an outcome by aggregating the information of the market. The trading part of the prediction market is what attracts the speculators who seek to "buy low and sell high". In prediction market, scoring rules play an important role in eliciting the true beliefs. So, the prediction market with a good scoring rule can often give accurate estimates of a probability of an outcome compared to the estimates given by the experts. For example, when compared to concurrent major opinion polls on U.S. presidential elections, the Iowa Electronic Market forecasts were more accurate 451 out of 596 times [BNR03]. Other such example is Hollywood Stock Exchange which is used to predict the award winners, it uses automated market makers to deal with the buying and selling of assets [Han03].

The prediction market are many times used for sensitive topics such as political and financial events (PredictIt), and even for science and technology forecasts (SciCast). Some prediction markets are even created for fun such as Hollywood Stock Exchange (`www.hsx.com`), by which there is no significant impact on the real world but people can enjoy, support their favourite actors and even benefit from the trading. So, here we have simulated one such prediction market for IPL showcasing how the inverstors belief towards a particular outcome can affect the prediction. Here, we have considered a question, "Will Mumbai Indian (MI) win the IPL 2020?". The prediction market is simulated with an automated market maker using a logarithmic market scoring rule (LMSR). The investors in this market are simulated using machine learning model.

1.1 Related work

Simple scoring rules are regularly used to elicit the probability estimates from the individuals. In simple scoring rule, a person reports a probability for each event, and gets paid depending on that report and the actual outcome of the event [Han07]. If the scoring rule maximizes the expected payoff of the agent when he reports his true belief, then such scoring rule is known as proper scoring rule. There are many proper scoring rules such as Quadratic, Spherical, Logarithmic, and Power Law. Logarithmic Scoring Rule is the only proper scoring rule in which the score of i^{th} event depends only on the report r_i and not on the probabilities given to the other events, i.e., r_j for $j \neq i$. To get a prediction of an event, one way is to collect the reports of the agents simultaneously by awarding each of them using the proper scoring rule. But the disadvantage of this method is, this method may be expensive as we might end up paying many times for the same prediction. Also, if the beliefs of the experts are different, then it is difficult to aggregate them. So, we want multiple agents to share their beliefs without paying each of them and need a single aggregated prediction. One mechanism to achieve this is Continuous Double Auction (CDA). In CDA, the market receives the sequence of orders. The orders are of two types: Limit Order and Market Order. In Limit Order, the agent posts the shares to the order book. In Market Order, the agent buys shares in the order book. But, CDA suffers from low market liquidity and huge spread problem. All these problems are overcome using prediction market mechanism using automated market maker proposed by Robin Hanson in his paper **Combinatorial Information Market Design (2003)**.

In Prediction Market mechanism with automated market maker, the agents make prediction one by one given the previous agents prediction. The Automated Market Maker first chooses a scoring rule (Market Scoring Rule) and initializes the cost according to its initial belief, and then the agents one by one update their beliefs by p. The agents who want to update their belief has to pay the cost according his report and the last report made. The agent can maximize his payoff only by stating his true belief. In this project, we have used Robin Hanson's Logarithmic Market Scoring Rule (LMSR). In this market scoring rule, the market maker's initial belief is uniform probability distribution. Hanson's LMSR is the only market scoring rule which allows conditional probability betting by preserving the marginal probabilities. Using LMSR, the maker's loss is bounded by $b * \log n$, where, n is the number of events. To exploit these advantages, we used LMSR in our project for simulating the market.

There are websites, apps which predict the winning team of a match using machine learning algorithms by taking past data as the training data for the algorithm. Another way of predicting the winning team is through prediction market. The existing prediction market simulations require the investors to specify the number of shares to trade in the market. In this project, we unburden the investors by calculating the number of shares to trade based on the reported belief of the investor towards the particular event.

1.2 Brief overview of the report

In Section 2, we introduce the scoring rule used which forms the base for this prediction market simulator. In Section 3, we have experimental results obtained after implementing prediction market simulator. In Section 4, we have simulation details explaining each module in our implementation. In Section 5, we finally conclude by giving the findings of our project and further possible future works.

2 Formal model of the problem

In order to find the stock price and trading cost, we are using Robin Hanson's **Logarithmic Market Scoring Rule (LMSR)** formulae in our Prediction market.

The LMSR formulae used are as stated below:

Computing Current Stock Price:

$$\text{Price}_i = \frac{e^{(q_i/b)}}{\sum_j e^{(q_j/b)}} \tag{1}$$

Cost Function:

$$\text{Cost} = b * ln\left(\sum_j e^{(q_j/b)}\right) \tag{2}$$

Trading Cost:

$$\begin{aligned}\text{Trading Cost} &= \text{Cost after Trade} - \text{Cost before Trade}\\ &= b * \left\{ ln\left(e^{((q_i+x)/b)} + \sum_{j\neq i} e^{(q_j/b)}\right) - ln\left(\sum_j e^{(q_j/b)}\right)\right\}\end{aligned} \tag{3}$$

Worst Case Loss of Market Maker:

$$\text{Loss} = b * ln\,(n) \tag{4}$$

where, q_i = number of shares of i^{th} stock
x = number of shares bought/sold
b is an arbitrary constant
n is the number of possible outcome of an event.

Though b is an arbitrary constant, it has significant effects in the market. If the value of b is small, the market price will change rapidly (i.e., even buying/selling a small number of shares will change the price a lot). On other hand, if the value of b is large, then the market price change will be very slow (i.e., large number of shares need to be bought/sold to change the price significantly).

So choosing a good value of b is important. We choose the value of based on the nature of the market. If the market liquidity is high, then the value of b should be high. If the market liquidity is low, the value of b should be low.

The value of b can also be determined by the market maker based on the worst case loss that he can afford.

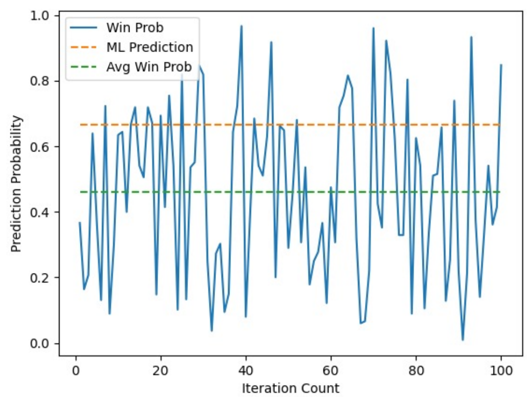
3 Main results/findings

The below table shows the experimental results of the prediction market simulation under different scenarios.

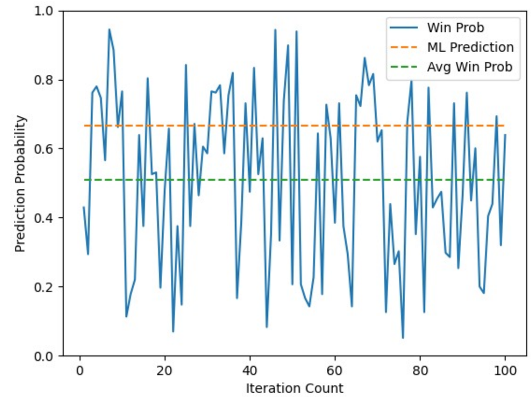
Case	Avg. Win Probability	Expert	Medium	Novice
1	0.4595181202204076	0	0	30
2	0.5102207017228415	6	0	24
3	0.5817573194745819	15	0	15
4	0.6993408980581859	24	0	6
5	0.7051583473007845	30	0	0
6	0.6510199817820458	10	10	10

Table 1: Experimental Results

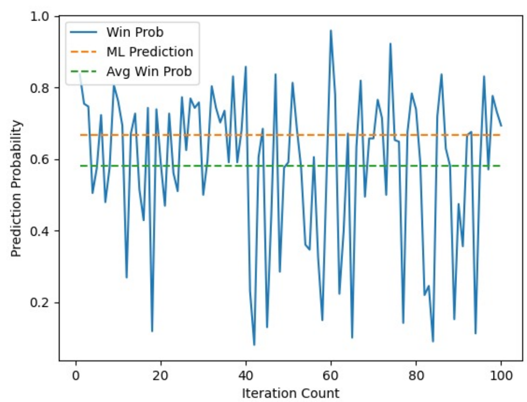
- **Case(1)** shows iteration wise change in win probability when there are only novice investors. As we can see in the figure, there is a large deviation in the win probability of each iteration from average value and ML model predicted value. Average value (nearly 0.45) is very less compared to ML model



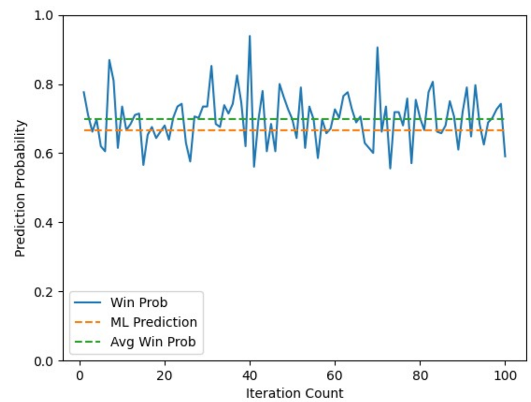
(a) Case: 1



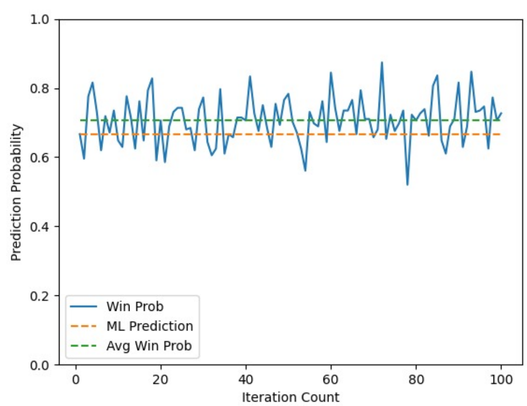
(b) Case: 2



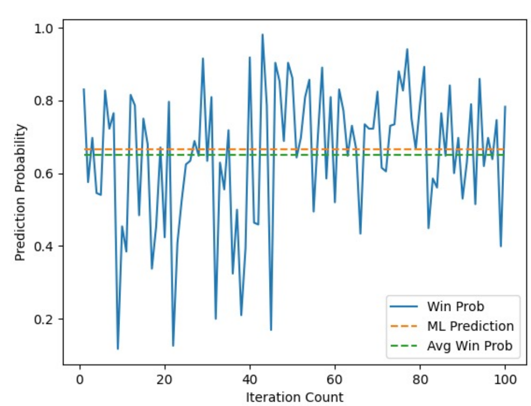
(c) Case: 3



(d) Case: 4



(e) Case: 5



(f) Case: 6

Figure 1: Experimental Results

predicted value (nearly 0.67). Also, the average value (0.45) which is less than 0.5 shows that they are betting on losing of the particular team, but the actual event was winning of team. So, the prediction by novice investors is very different from the actual outcome.

- **Case(2)** shows iteration wise change in win probability when there are some expert investors (20% experts) along with novice investors. As we can see in the figure, the deviation in this case is similar to the above case. But, the average value (0.51) is more compared to above case average value (0.45) and it is close to the ML predicted value (0.67) than above case. Also average value (0.51) which is slightly above than 0.5 shows that expert investor tried to move average value towards actual event(win of team) as compared above case.

- **Case(3)** shows iteration wise change in win probability when there are 50% expert investors and 50% novice investors. Compared to the first two cases, average value of this case (0.58) is near to ML predicated value (0.67). As more expert investors are betting, the average value is moving towards ML predicted value (0.67) and actual event (win).

- **Case(4)** shows iteration wise change in win probability when there are some novice investors (20% experts) along with expert investors. From this case we can observe that, as the no. of expert investor increases, the deviation from average value and ML predicted value decreases. Average value (0.69) is near to ML predicted value (0.67). And average value (0.69) shows that team will win and actual event is also win.

- **Case(5)** shows iteration wise change in win probability when there are only expert investors. Here, the deviation from the average value and ML predicted value is very less compared to all cases. And average value (0.70) is indicating for win of team and actual event is also win.

- **Case(6)** shows iteration wise change in win probability when there are expert, medium and novice investors in same proportion. Here, the deviation is there due to novice and some medium investors but also some medium investors and expert investors are keeping average value (0.65) near to ML predicted value (0.67).

From above cases we can say that the variation in win probability not only depend on factors affecting the actual event, but also on the type of investors (only experts; only novices; mixture of experts, medium, novices) betting in the prediction market.

Multiple Trading Per Investor:

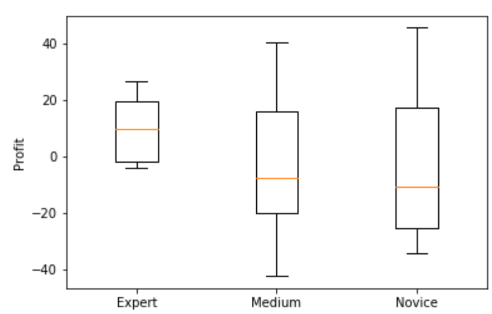


Figure 2: Profit of investors

The box plot in the Figure 2 shows the variation in the profits of different type of investors (Expert, Medium and Novice) after allowing them to trade for 50 iterations. In each iteration, same investors randomly trade, according to the current win probability predicted by the market, based on their initial beliefs. From the figure, we can interpret that the loss of the expert investors is negligible, while that of medium-knowledge and novice investors is significant. The orange line in the box plot represents the median loss of each type of investor. The expert investors median line is above the origin representing that many expert investors are making profit, while that of other two type of investors is below the origin representing that many investors are incurring a loss.

4 Experiments/Simulations

The Simulation of Prediction Market for IPL is done in three steps:

- **Simulation of Investors' Belief**

  For this experiment, three types of investors are simulated: Expert, Medium, Novice. The investor's belief regarding a particular event is obtained via simulation as mentioned below:

  The Machine learning model is used to predict the winning probability of a particular team. After trying out various multi-class classification models, we came to a conclusion that the **Decision Tree** model gave the good result of prediction compared to other models. The database used for ML model consists of two features: Win Ratio and Position on Points Table after the league stage, and one label: 0, 1 or 2 (0: Loser, 1: Winner, 2: Runner-Up).

  - **Expert Investor:** The prediction probability for expert inverstors is simulated by introducing a variation of 0.1 to the probability predicted by the ML model.
  - **Medium Investor:** The prediction probability for medium inverstors is simulated by introducing a variation of 0.25 to the probability predicted by the ML model.
  - **Novice Investor:** The prediction probability for novice investors is generated randomly using random number generator.

- **Buying and Selling of Shares**

  The buying and selling of shares based on the prediction probability submitted by the investor is done by the following formula:

  - If predicted probability by investor is greater than the current market predicted probability, then number of shares to be bought are calculated as follows:

$$\text{No. of shares} = \left(\frac{\text{Current probability - Predicted probability}}{1.01 - \text{Predicted probability}}\right) * 10 \tag{5}$$

  - If predicted probability by investor is less than the current market predicted probability, then number of shares to be sold are calculated as follows:

$$\text{No. of shares} = \left(\frac{\text{Current probability - Predicted probability}}{0.01 + \text{Predicted probability}}\right) * 10 \tag{6}$$

  Next, in order to find the current market price of the stock and trading cost to perform a transaction, Robin Hanson's **Logarithmic Market Scoring Rule (LMSR)** formulae are used. The LMSR formulae used are as stated below:

  - **Computing Current Stock Price**:

$$\text{Price}_{\text{win\_share}} = \frac{e^{\text{win\_share/b}}}{e^{\text{win\_share/b}} + e^{\text{lose\_share/b}}} \tag{7}$$

– **Cost Function**:

$$\text{Cost} = b * ln(e^{\text{win\_share/b}} + e^{\text{lose\_share/b}}) \tag{8}$$

– **Trading Cost**:

$$\text{Trading Cost} = b * ln\left(\frac{e^{\text{win\_share/b + x}} + e^{\text{lose\_share/b}}}{e^{\text{win\_share/b}} + e^{\text{lose\_share/b}}}\right) \tag{9}$$

– **Worst Case Loss of Market Maker:**

$$\text{Loss} = b * ln(2) \tag{10}$$

- **Market Resolution**

  After the declaration of final outcome, the market is resolved by paying Rs.1/share to the investors holding the shares of the final outcome, i.e., if the final outcome is win, then the investors holding win shares (i.e., the current holding is positive) are payed Rs.1/share and likewise if the final outcome is lose, then the inverstors holding lose shares (i.e., the current holding is negative) are payed Rs.1/share.

5 Summary and Discussions

- To conclude we can say that the prediction market is simulated by obtaining the investors beliefs towards a particular event. The market's prediction of the event changes constantly on acquiring the belief of individual investors.
- After the market is closed, it is resolved based on the final outcome of the event. If the outcome is in favour of the users belief, then he will make profit, else he will incur a loss.
- The market maker will make a profit if the prediction after crowdsourcing is against the final outcome, else it will incur a loss.
- If the investors are allowed to predict only once, then the profit and loss of each type of investor depends on the sequence in which the predictions are made, i.e., are expert investors investing at the end or the novice investors? But, when the investors are allowed to trade multiple times, the expert investors incur less loss compared to the other two type of investors.

Possible Future Works

- The prediction market is generated for "Will MI(particular team) will win IPL 2020?". Possible outcomes of this question are: Win or Lose. Can change it simulate question "Which team will win IPL 2020?" (Possible outcomes of this question are: All teams participating in IPL 2020).
- In this simulation prediction market is opened after the league stages of IPL over. Can convert it to prediction market is opened with very first match of IPL. So win probability of different team will vary match by match.
- ML model is predicting just based on numbers of matches team won in league stage and position on points table after league stage completed. But can consider factors that how different investors performed in league stage matches, how closely they won or lost league stage matches, pitch condition, dew factor, toss etc. If we can accumulate the data-set consisting above factors then ML prediction will much closer to actual outcome.

References

[BNR03] Joyce Berg, Forrest Nelson, and Thomas Rietz. "Accuracy and forecast standard error of prediction markets". In: *Tippie College of Business Administration, University of Iowa* (2003).

[Han03] Robin Hanson. "Combinatorial information market design". In: *Information Systems Frontiers* 5.1 (2003), pp. 107–119.

[Han07] Robin Hanson. "Logarithmic markets coring rules for modular combinatorial information aggregation". In: *The Journal of Prediction Markets* 1.1 (2007), pp. 3–15.