Quant Guild Report

ME23B064
Rohit B

Quantifying Me:

- I am Rohit from the Department of Mechanical Engineering. I am very interested in probability and enjoy learning more about it. I love solving brainteasers and logical puzzles. I also enjoy competitive coding and take part in contests on a regular basis. I am also fairly good in speed math. In my free time I play chess and basketball.
- The motivation for me to join the guild is to put myself into a lot of interesting problems, meet like minded people and learning from them. I want to learn more about quantitative finance, how we can view it from a mathematical perspective and possibly represent IIT Madras in the upcoming Inter IIT Teach Meet:)
- I don't have any active commitments at the moment but I have applied for a YRF project under Prof.Rakhi Singh on developing variable selection methods for mixed inputs, the interview nor the selection is not done yet. The other thing is I am a part of Team Abhiyaan but my tenure as a junior is over now and the next set of juniors have been recruited.
- I don't have much expireince in quant. In fact I didn't know what quant was until the guild recruitment of last year. When I went through applications of all guild, I found the problem statement of quant guild very interesting . I did the last year's trading bots and the kaggle competition, that's my only prior experience.

General Idea

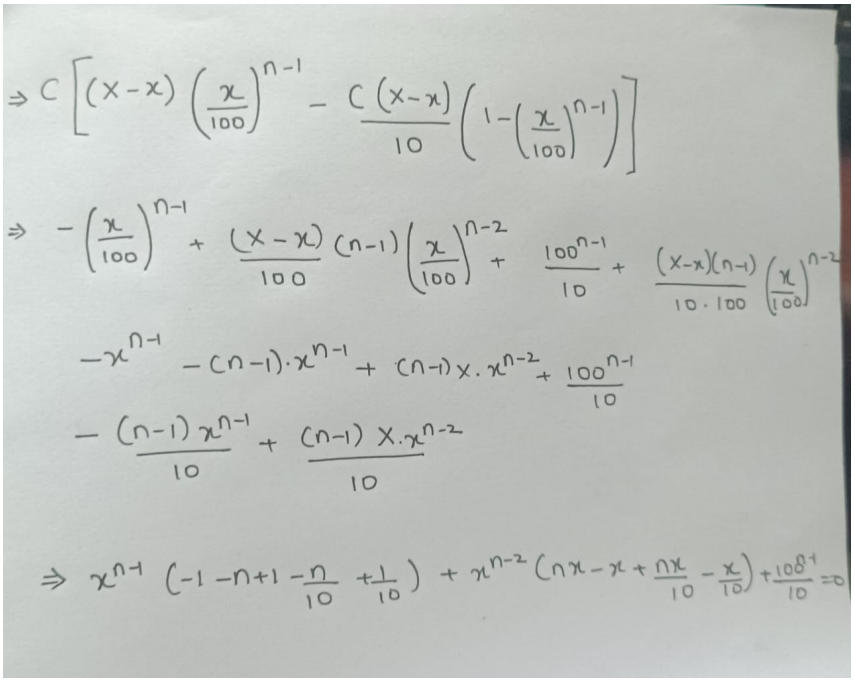
I tried writing the expected returns for each case in terms of the known variables and the unknown optimal bid. Then i differentiated this expression with respect to the optimal bid and set it to zero to find the optimal bid for each variant. In all cases I am assuming opponents' bids to be a uniform random variable which lies in the range .[0,100].

Variant 2:

I am starting with variant 2 because I think this is a simpler variant. Its because I believe we don't have to consider the best winning bids of previous rounds for this. The optimal bid will only depend on the random X_i we get at each round. The expected profit in terms of confidence interval(c), random X_i, number of bidders(n) and unknown bid x is

$$C \cdot (X - x) \cdot \left(\frac{x}{100}\right)^{n-1} - \frac{C \cdot (X - x)}{10} \cdot \left(1 - \frac{x^{n-1}}{100^{n-1}}\right)$$

We observe that we can take C common, and if the inside term is positive, C should be 1 and if the inside term is negative its better for us not to bid anything. Differentiating and simplifying the equation:



$$x^{n-1}(1 - 11n) + x^{n-2}(11nx - 11x) + 100^{n-1} = 0$$

Then I used numerical method in python to find the roots of this equation. We also need to verify whether our expected payoff is positive else we can also choose not to participate in this round. The condition to check this simplifies to

$$x > \frac{100}{11^{\frac{1}{n-1}}}$$

We bid in a particular round if we find a root which satisfies this condition.

Variant 1:

I initially tried going with the nash equilibrium case for this in mock auction, that is assuming everyone bids ideally maximizing their payoff but this didn't work out. To maximize payoff we need to multiply expected profit by the probability we come either first or second, but as players know the winning bids and also X in this case is the maximum of X of all players it can be estimated with a good probability, the first strategy didn't work out. So in this case we just need to maximise our chance of winning every time. To do this first I calculated expected maximum of of all n random variables X given one of the random variable takes the value of the X we got. This is done by calculating CDF then differentiating to find the PDF. After that we can find expected value by doing a summation for each value.

$$E[\max(X_1, \ldots, X_n) \mid X_1 = x] = x \cdot \left(\frac{x+1}{101}\right)^{n-1} + \sum_{y=x+1}^{100} y \cdot \left[\left(\frac{y+1}{101}\right)^{n-1} - \left(\frac{y}{101}\right)^{n-1}\right]$$

After this I took average of the winning bids from last rounds. Now my idea was to bid higher than the average bid but lower than expected value of X to get profit. Also I wanted my bid to be a bit above the average winning bid and not too much to increase profits as much as possible. To do this I bid an amount which was weighed higher towards the winning bid side. I experimented with multiple weighing methods and finally decided to weigh in the ration of n:1. I also added a safe bid when capital goes too low.

Variant 3:

I found this variant to be tricky and made an assumption to simplify the model.I assumed that the difference between the highest and second highest bid is lower than X-bid. I assumed this because people try to use data from the winning bids, so everyone's bid should be much closer than X-bid. I calculated average winning bids- average second bids and stored it in a variable k. We can also use the same expected value formula from variant 2 to calculate expected X_max here also. So our expected value expected value equation is

$$E = \left(\frac{x}{100}\right)^{n-1} (X_{\max} - x - k) - \frac{n-1}{2} \left(\frac{x}{100}\right)^{n-2} \left(\frac{100-x}{100}\right) (X_{\max} - x - k)$$

Then I differentiated this equation and found roots using the sympy library in python. I feet this variant of the auction would give a much more volatile result compared to variants 1 and 2.

Conclusion and Possible Improvements:

I think maximizing expected payoff by differentiating is a solid and reliable strategy, especially for variant 2. The one strategy which I think can outsmart my bot is for variants 1 and 3 there could be a better way to do something with the data of winners of previous rounds. I have only taken averages. I feel something better can be done with that information for slight improvements.