A Probabilistic Agent with Memory

Jonathan Stokes

August 27, 2024

1 Introduction

The goal here is to create a simple GOPS agent based on two components:

- Opponent Bid Prediction
- Bid Efficiency

Opponent bid prediction attempts to predict the bid an opponent is going the make given GOPS prize r. Bid efficiency is a measure of how efficient an agent bid is given prize r and the opponent bid prediction $\hat{b}_o$. The goal of the agent is to predict the opponents bid $\hat{b}_o$ and use this prediction to compute the maximum bid efficiency over the set of their cards, then select the bid card b_e^* which maximizes the bid efficiency.

In other words, given $(r, \hat{b}_o)$ the agent selects their bid using the function,

$$b_e^* = \underset{b_p \in \mathcal{H}_p}{\text{argmax}}(e_{agent}(b_p, \hat{b}_o, r)) \tag{1}$$

where e_{agent} is the bid efficiency function, $\mathcal{H}_p$ is the set of cards in the agents hand and b_p is one of these cards, $\hat{b}_o$ is the predicted opponent bid, and r is the current GOPS prize.

2 Notation

Let r be the prize value of a round of GOPS and t the round. Let b be a generic bid, $\hat{b}$ be a generic predicted bid, b_p and b_o be the agent and opponent bids respectively, $\hat{b}_o$ be the predicted opponent bid, and b_e^* be the predicted maximum efficiency agent bid. Note it is not necessarily the case that $b_p = b_e^*$, this distinction is useful in explaining the efficiency function. Let $s(r, t)$ be a bid strategy conditioned on prize r and round t, and $\mathcal{S}$ a set of these strategies, and $\hat{s}(r, t)$ be a predicted strategy. Additionally let $\mathcal{H}$ be a hand of cards, $\mathcal{H}_p$ and $\mathcal{H}_o$ the agent and opponent hands respectively.

3 Bid Prediction

I would guess that there are existing solutions for predict opponent bids, but finding, understanding, and adapting those solutions to this case and evaluating them is more work than I want to get into at the moment. So lets create an ad-hoc algorithm for this.

First lets define a bid strategy conditioned on the prize value r and round t as mapping $s(r, t) \to b$ where b is the bid value selected using strategy $s(r, t)$. Now lets define a set of bid strategies for all prizes r and turns t as,

$$\mathcal{S} := \{s(r, t) | 0 \leq r \leq 13, 1 \leq t \leq 13\} \tag{2}$$

Additionally, lets initialize $\mathcal{S}$ for $t = 0$ with $s(r, 0) = r$ for $1 \leq r \leq 13$. This initialization for $t = 0$ sets the agents bid prediction prior handling the cold start problem for $t = 1$ by assuming on turn $t = 1$ the opponents strategy is to bid the the prize value r.

If on turn $t = 1$ the opponent uses strategy $s(r, 1)$ add this strategy to the set bid strategies $\mathcal{S}$ such that,

$$\mathcal{S} = \mathcal{S} \cup \{s(r, 1)\} \tag{3}$$

Continue adding observed opponent strategies to the set $\mathcal{S}$ in this manner each turn.

Now if c is the current turn and q the current prize how are we going to form a prediction of the opponents strategy $\hat{s}(q, c)$, where $\hat{s}(q, c) \rightarrow \hat{b}$ is the predicted bid? Probability, my dear Watson. The opponents predicted strategy $\hat{s}(q, c)$ is selected from the set of strategies $\mathcal{S}$. The probability of selecting a strategy $s(r, t) \in \mathcal{S}$ as the predicted opponent strategy $\hat{s}(q, c)$ is,

$$p_s(c, r, t, q) = \frac{\mathbb{1}\{s(r, t) \in \mathcal{S}\} e^{-(c-t)-|q-r|}}{\sum_{a=1}^{13} \sum_{b=0}^{c-1} \mathbb{1}\{s(a, b) \in \mathcal{S}\} e^{-(c-b)-|q-a|}} \tag{4}$$

The result of selecting strategies using this probability distribution is that strategies $s(r, t)$ where (r, t) are closer to the current turn c and prize q are more likely to be selected as $\hat{s}(q, c)$.

Having selected $\hat{s}(q, c)$ the predicted bid is determined by the mapping $\hat{s}(q, c) \rightarrow \hat{b}$. However b^* may not be in the opponents hands, therefore let the predicted opponent bid be,

$$\hat{b}_o = \underset{b \in \mathcal{H}_o}{\operatorname{argmin}}(|b - \hat{b}|) \tag{5}$$

with ties broken randomly.

4 Bid Efficiency

There is probably a provably optimal measure of bid efficiency but again for the moment I will use an ad-hoc notion of bid efficiency. This notion of bid efficiency is a piecewise-defined function that depends on the players bid, opponents bid and the prize value, (b_p, b_o, r).

Bid efficiency is defined as,

$$e_{agent}(b_p, b_o, r) = \begin{cases} \frac{rc}{\log_2(b_p+1-b_o)} & \text{for } b_p > b_o \\ 0 & \text{for } b_p = b_o \\ b_o - b_p & \text{for } b_p < b_o \end{cases} \tag{6}$$

where b_p is the player bid, b_o is the opponent bid, and r is the prize value. The parameter c determines the value of a bid larger than b_o. I solved for c under the assumption that the agent should bid low for any prize value under 7. To ensure this I solved for c such that,

$$b_o - min(b_p) < \frac{cr}{log_2(1 + b_p + b_o)} \tag{7}$$

where $min(b_p)$ is the minimum possible player bid or 1 for $1 \leq b_p \leq 13$. Rearranging this gives,

$$\frac{(b_o - min(b_p))log_2(1 + b_p + b_o)}{r} < c \tag{8}$$

Letting $r = 7$, $b_o = 7$, $b_p = 8$ such that $b_p > b_o$, and $min(b_p) = 1$ gives,

$$\frac{6log_2(16)}{7} = 3.429 < c \tag{9}$$

Setting $c = 3.5$. See Figure 1 and Figure 2 for plots e_{agent} under a number of different parameters.

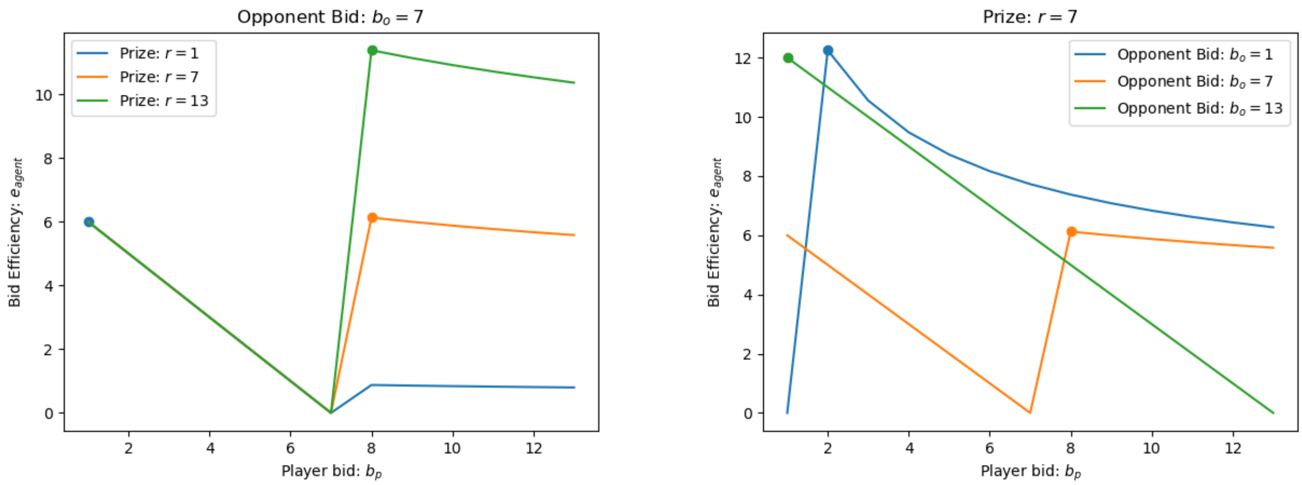

Figure 1: e_{agent}: ●'s mark the maximum values of the functions.

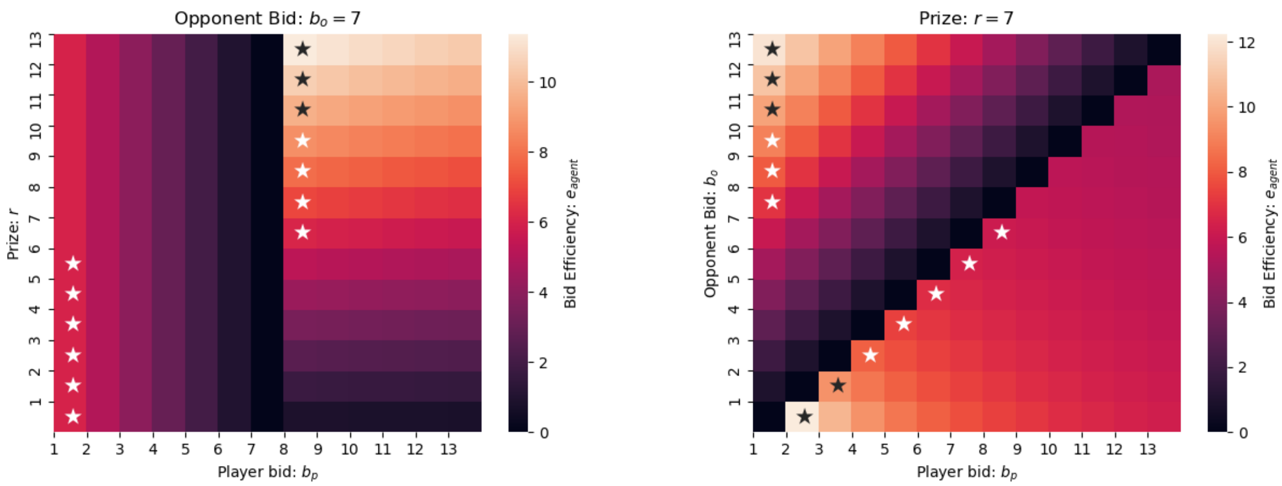

Figure 2: e_{agent} heatmap where the ★'s indicate the optimal player bid for each row.

While I think these figures do a good job of characterizing e_{agent} I am not sure there is much of a takeaway from these figures as e_{agemt} is an ad-hoc function.

5 Agent Performance

I evaluated the probabilistic agent with memory (ProbMem) against a uniformly random agent (Random) and an agent which attempts to always play the prize card (PrizeCard). Table 1 gives the fraction of player wins for 100k GOPS games.

		Opponent		
		Random	PrizeCard	ProbMem
Player	Random		0.065	0.402
	PrizeCard	**0.935**		0.011
	ProbMem	**0.594**	**0.989**	

Table 1: The bold numbers represent wining player strategies against opponent strategies.

One interesting take away from the results in Table 1 is that while the PrizeCard strategy performs very well against the Random strategy, the ProbMem strategy performs well against both the Random and the PrizeCard strategy.

6 Conclusion

I had the idea for a probabilistic agent with memory while thinking about training Transformer models to play GOPS. I thought that I could use it to train a Transformer model or as a baseline to evaluate Transformer models against and hopefully I will get to try this out in the future. But what surprised me is that although the probabilistic agent with memory wins less then 60% percent of the time against a random agent it feels competitive when playing against it as a human. The probabilistic agent with memory makes bids that seem rational to a human player since those bids are made based knowledge of the human players previous bids.

This implies that evaluating agent performance against a random agent is likely a bad measure of how “good” a GOPS agent is. Ideally the agents performance would be evaluated against thousands of human games, but given that this is not currently feasible given time and financial constraints, perhaps I should evaluate agents against a basket of opponent agents. The competitive feel of a GOPS agent may have more to do with its flexibility in adapting to different strategies then to a 90% plus win rate against any given strategy. Of course this opens the non-trivial problem of selecting and weighting a basket of opponent strategies to evaluate GOPS agents against.