Math 231 Project: Riding the Bus

Harry Sha (harry2)

November 19, 2020

Abstract

In this paper we explore the mathematics of *riding the bus*, a popular card guessing game. In particular, we aim to compute and estimate the expected number of guesses and failures under different variants of the game and play strategies. We also aim to understand the distribution of guesses and failures in idealized models, and see how well these models correspond to the real world.

1 Introduction: riding the bus

Rules and variants. Riding the bus is a card guessing game with simple rules. The player must make 4 correct guesses in a row and starts over from question 0 at any incorrect guess.

- **Question 0**: red or black? (Is the color of the next card red or black?)
- **Question 1**: higher or lower? (Is the value of the next card higher or lower than the original card?)
- **Question 2**: inside or outside? (Is the value of the next card in between or outside of the previous two cards?)
- **Question 3**: suit? (What is the suit of the next card?)

When the deck runs out of cards, we reshuffle and continue. In the real world, *riding the bus* is typically played as a drinking game where the cost of an incorrect guess is a sip of whatever drink you have in your hand. Therefore, this research will help determine the risks involved in playing the game, and understand the expected monetary cost of playing the game. We will study two variants of the game.

- **Lenient.** In this version of the game, higher, lower, inside and outside are defined leniently in that a card is considered both higher and lower than itself.
- **Harsh.** In this version of the game, higher, lower, inside and outside are defined harshly in that a card is considered neither higher nor lower than itself.

In this work, we consider a generalized deck with n values and m suits. A deck with n values and m suits is modeled as $\{0, 1, ..., n-1\} \times \{0, 1, ..., m-1\}$.

Strategies: Define a strategy as a (possibly randomized) algorithm, φ that takes in a history of the game so far and outputs a guess. The following are some strategies that we will analyse:

1. φ_{drunk}. Select a random guess.
2. φ_{greedy}. Selects the guess that maximizes the probability of getting the current question correctly without using memory.
3. $\varphi_{\text{hyperthymestic}}$. Uses the same algorithm as the greedy player, except it remembers all the seen cards, and and hence knows the distribution of remaining cards in the deck.

This work. To our knowledge, no significant mathematical or statistical analyses have been conducted on riding the bus. Thus, this paper aims to answer fundamental questions related this game. At a high level, we are interested in the expected number of drinks and guesses in the game, the optimality of the greedy strategy, and the edge that can be acheived if one memorizes the cards.

2 Infinite deck model

2.1 Markov chain

In this model, we view the deck as infinitely long. The probability of drawing any card at any point in the game is $\frac{1}{mn}$. This model significantly simplifies the calculations. We can model the game played by the greedy player (note that the hyperthymestic player has no advantage in this model) as a discrete time Markov chain with the following states:

- s_0, the state prior to answer question 0.
- $s_{1,k}$, the state prior to answering question 1, and seeing a card with value k.
- $s_{2,k}$, the state prior to answering question 2, where the absolute value of the difference of the two prior cards is k.
- s_3, the state prior to answering question 3.
- s_4, the state where the game won.

Note that the strategies that we are interested in only use the value of the first card in their guess for question 1 and the absolute difference of the the value of the first two cards in the guess for question 2, so these states fully capture the game. The Markov chain for $n = 4$ in the lenient variant is show in Figure 1. Of note here is that the subgraph involving the transitions from question 1 to question 2 are not a complete bipartite graph. In particular, we see that there is no transition from $s_{1,1}$ to $s_{2,3}$, as no value that is drawn from $\{0, 1, 2, 3\}$ has an absolute different of 3 with 1.

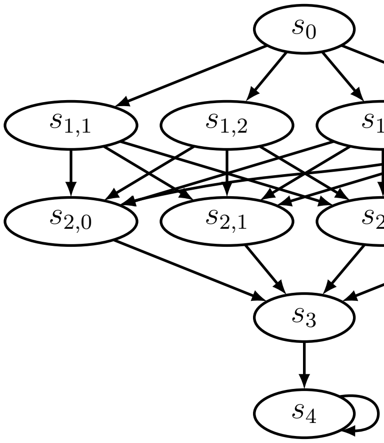


Figure 1: Markov chain for $n = 4$. Note that transitions back to s_0 are omitted.

2.2 Transition probabilities

Let $T(v, x, y)$ be the probability of transitioning from state x to state y in the v variant of the game for the Markov chain described in the previous section. We'll calculate these question by question.

Quesion 0. Note that regardless of the value of n, m, and v we have that the probability that we guess the color of a card incorrectly is 1/2, and thus $T(v, s_0, s_0) = 1/2$. If we guess the color correctly, it is is equally likely that the card has any of then n values. Therefore for all $k \in \{0, 1, ..., n-1\}$, we have $T(v, s_0, s_{1,k}) = \frac{1}{2n}$.

Quesion 1. Having seen value k after the first draw, the greedy player must then guess whether or not the next card was higher or lower. Under the lenient rules, there are $k+1$ cards '*lower*' than k and $n-k$ cards '*higher*' than k. Under, the harsh rules, there are k cards '*lower*' than k and $n-k-1$ cards '*higher*' than k. Let $R_1(\text{lenient}, k) = \max\{k+1, n-k\}$, and $R_1(\text{harsh}, k) = \max(k, n-k-1)$. The greedy player guesses wrong if any of $R_1(v, k)$ cards are drawn, so $T(v, s_{1,k}, s_{0,0}) = 1 - \frac{R_1(v,k)}{n}$. In the lenient version, if we guess correctly, we know that the absolute difference between the two cards is less than $R_1(\text{lenient}, k)$. In the harsh version, if we guess correctly, we know that the absolute difference between the two cards is between 1 and $R_1(\text{harsh}, k)$. Therefore,

$$T(\text{lenient}, s_{1,k}, s_{2,j}) = \begin{cases} 1/n & = 0 \leq j < R_1(k) \\ 0 & \text{otherwise} \end{cases},$$

and

$$T(\text{harsh}, s_{1,k}, s_{2,j}) = \begin{cases} 1/n & 0 < j \leq R_1(k) \\ 0 & \text{otherwise} \end{cases},$$

for all $j, k \in \{0, 1, ..., n-1\}$.

Question 2. If the difference between the first two cards is k, then under the lenient rules, there are $k+1$ numbers '*inside*' them, and $\min(n-k+1, n)$ numbers '*outside*' them, where we need to take the min with n to account for the case where $k = 0$, and under the harsh rules, there are $k-1$ '*inside*' and $n-k-1$ numbers '*outside*'. Let $R_2(\text{lenient}, k) = \max(k+1, \min(n-k+1, n))$, and $R_2(\text{harsh}, k) = \max(k-1, n-k-1)$, this is the number of cards that let the player move to the third question. Therefore, $T(v, s_{2,k}, s_{0,0}) = 1 - R_2(v, k)/n$, and $T(v, s_{2,k}, s_3) = R_2(v, k)/n$.

Question 3. We get the final question correct with probability $1/m$, so $T(s_3, s_4) = 1/m$, and $T(s_3, s_0) = 1 - 1/m$

We are done once we get to Q_4, so $T(s_4, s_4) = 1$. All transition probabilities that were not mentioned are have probability 0.

View T_v as the transition matrix for the Markov chain, with transition probabilities defined as above, and e_x be the vector corresponding state x (1 at the index corresponding to state x, and 0 everywhere else). Let $D(v, t, x) = e_x(T_v)^t$ be the distribution of states after t time steps starting from state x, and let $P(v, t, x, y) = D(v, t, x)[y]$ be the probability that we are in state y at time t where we start from state x.

2.3 Number of drinks

Let $N_d(v)$ be the number of drinks in a game played by the greedy player in a variant v under the infinite deck model. In this section, describe the distribution of $N_d(v)$ through an analysis of the Markov chain.

Theorem 2.1. *Let*

$$p_{lenient} = \frac{1}{2mn^3}\left(\sum_{k=0}^{n-1} \min(n, 2(n-k)) \cdot \max(k+1, \min(n-k+1, n))\right),$$

and

$$p_{harsh} = \frac{1}{2mn^3}\left(\sum_{k=1}^{n-1} \min(n, 2(n-k)) \cdot \max(k-1, n-k-1)\right).$$

Then we have $N_d(v) \sim Geom(p_v) - 1$

Proof. The game ends when we successfully make 4 correct guesses in a row. This occurs with probability $P(v, 4, s_0, s_4)$. Therefore, the number of tries required to win the game is distributed as $\text{Geom}(P(v, 4, s_0, s_4))$. The number of drinks is then the number of tries minus 1 (we don't drink after winning). Thus it suffices to show that $P(v, 4, s_0, s_4) = p_v$. We'll show the theorem for the harsh variant (the lenient variant is similar). Expanding the calculation for $P(v, 4, s_0, s_4)$, we get that

$$\begin{aligned}
P(\text{harsh}, 4, s_0, s_4) &= \frac{1}{4} P(\text{harsh}, 3, s_0, s_3) \\
&= \frac{1}{m} \sum_{k=1}^{n-1} T(\text{harsh}, s_{2,k}, s_3) \cdot P(\text{harsh}, 2, s_0, s_{2,k}) \\
&= \frac{1}{mn} \sum_{k=1}^{n-1} R_2(\text{harsh}, k) \sum_{j=0}^{n-1} T(\text{harsh}, s_{1,j}, s_{2,k}) \cdot P(\text{harsh}, 1, s_0, s_{1,k}) \\
&= \frac{1}{2mn^2} \sum_{k=1}^{n-1} R_2(\text{harsh}, k) \cdot \frac{\deg^-(s_{2k})}{n} \\
&= \frac{1}{2mn^3} \sum_{k=1}^{n-1} \max(k-1, n-k-1) \cdot \min(2(n-k), n) \\
&= p_{\text{harsh}},
\end{aligned}$$

where we get the 4th equality because $T(\text{harsh}, s_{1,j}, s_{2,k})$ is $1/n$ for each j that is able to transition to $s_{2,k}$ and 0 otherwize. Thus $\sum_{j=0}^{n-1} T(\text{harsh}, s_{1,j}, s_{2,k})$ is the indegree of $s_{2,k}$ in the Markov chain divided by n.

□

For a standard deck, we can plug in the $n = 13$, and $m = 4$, to get $p_{\text{lenient}} \approx 0.0793$, and $p_{\text{harsh}} \approx 0.056$. Thus we can compute $E[N_d(\text{lenient})] \approx 11.6$, and $E[N_d(\text{harsh})] \approx 16.8$. Thus we can expect to drink roughly 5 more times per game when playing the greedy strategy with the harsh rules compared to the lenient rules. The medians of the lenient and harsh variants are $\lceil -1/\log_2(1 - p_{\text{lenient}}) \rceil - 1 = 8$, and $\lceil -1/\log_2(1 - p_{\text{harsh}}) \rceil - 1 = 13$ respectively and the mode of both distributions is 0. These statistics conform pretty well to my experience playing this game as I have often been surprised by being able to get all 4 guesses on the first try, yet other times the game would go on for a long time.

An interesting implication of our theorem is that asymptotically the number of drinks depends only on the number suits in a deck and not the number of values in the deck.

Remark 2.1. *Let n and m be the number of values and suits in the deck. Then $E[N_d(v)] \in \Theta(m)$.*

Proof. Note that it suffices to show that the $p_v \in \Theta(\frac{1}{m})$. To do so, we'll show that the numerator in our expression for p_v is $\Theta(n^3)$. Clearly, in either variant, the numerator is $O(n^3)$ as we are summing n terms that are each at most $n \cdot n$. Then it suffices to show that the numerator in the harsh variant is $\Omega(n^3)$.

Note that $\max(k-1, n-k-1) \geq n/2 - 1$, so the numerator is at least

$$\Omega(n) \sum_{k=1}^{n-1} \min(n, 2(n-k)) \geq \Omega(n) \sum_{k=1}^{n-1} k \in \Omega(n^3)$$

as required.

□

An intuitive explanation of this result is that for large n, we have that picking a value from the deck approaches sampling from a uniform distribution, which does not depend on n, and that in order to win the game, we need to guess the final suit which requires $\Theta(m)$ tries on expectation. Table 1 shows $E[N_d]$ for a selection of parameters.

	Variant	n	m	$E[N_d]$	Probability of winning on first try
0	lenient	13	4	11.6	0.079
1	lenient	13	10	30.5	0.031
2	lenient	13	20	62.0	0.015
3	lenient	20	4	12.2	0.075
4	lenient	50	4	13.1	0.070
5	harsh	13	4	16.8	0.056
6	harsh	13	10	43.6	0.022
7	harsh	13	20	88.2	0.011
8	harsh	20	4	15.6	0.060
9	harsh	50	4	14.4	0.064

Table 1: Expected number of drinks and winning probabilities for different variants and parameters of the game.

2.4 Number of Guesses

Another quantity that we are interested in is the number of guesses in the game, $N_g(v)$, which can be seen as some sort of approximation to the time spent playing the game.

Figure 2 shows a visualization of the distribution of states in the Markov chain at selected time steps, and Figure 3 shows the probability of being at each question at each time step. We see that being in the final state is the most likely state by $t \approx 11$, and that the greedy player has won by $t \approx 20$ with over 50% probability.

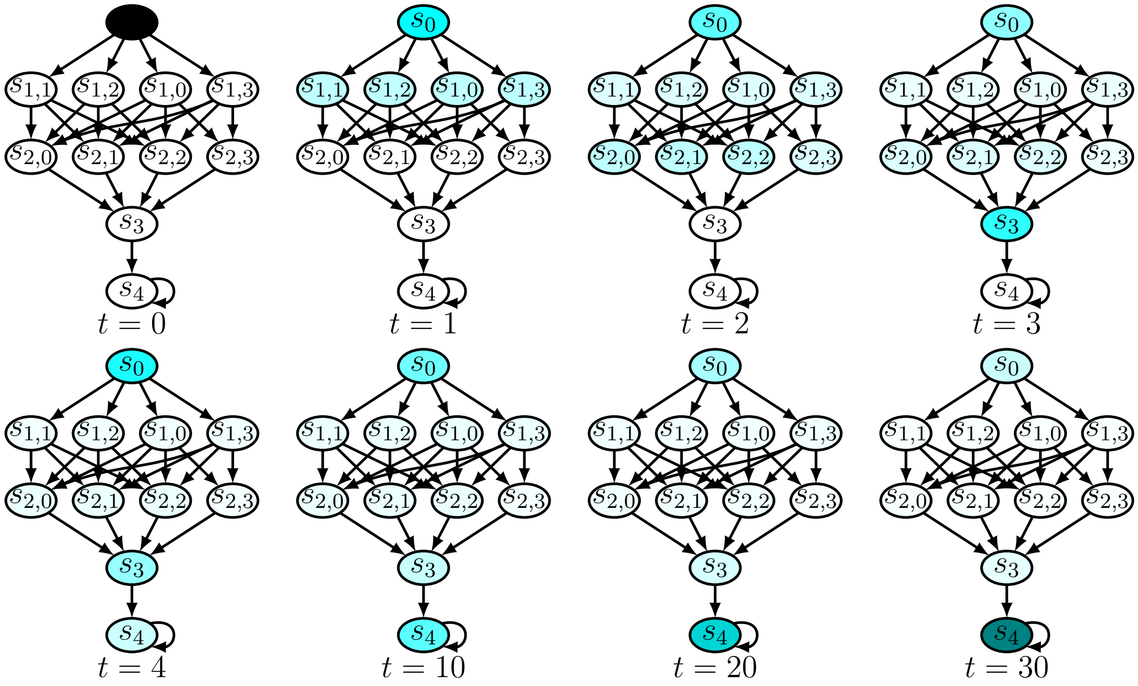
Table 2 show statistics related to the distribution of failed guesses for the greedy strategy and the lenient variant based on a simulation of 100,000 games. In particular, we find that $E[N_g(\text{lenient})] \approx 27.9$. An interesting result from this simulation is that we are actually likely to answer questions 1 and 2 when we get there (79%, and 82% respectively), and are expected to fail on those questions only about once per game.

2.5 Optimality of the greedy strategy

Intuitively, the greedy strategy is pretty good - any deviation and we incur a higher risk getting the current question wrong and starting over. However, is this an optimal

	fail of q0	fail on q1	fail on q2	fail on q3	N_g
Probability	0.50	0.79	0.81	0.25	-
Mean	6.3	1.3	1.0	3.0	27.9

Table 2: Distribution of failed guesses for the greedy strategy in the lenient variant under the infinite deck model.



$t = 0$ $t = 1$ $t = 2$ $t = 3$

$t = 4$ $t = 10$ $t = 20$ $t = 30$

Figure 2: Distribution of states at time step t for $n = 4$, $m = 4$. Darker means higher probability

strategy? One result that suggests that it might be possible to do better than the greedy strategy is that given a uniform distribution on pairs of cards in the standard (infinite deck), the probability that we answer question 2 correctly ('inside or outside') under the harsh rules is

$$\frac{1}{13^2}\sum_{i=0}^{12}\sum_{j=0}^{12} T(\text{harsh}, s_{2,|i-j|}, s_3) \approx 0.675.$$

However, given the distribution defined by the markov chain and the greedy strategy, we have that the probability that we answer question 2 correctly is

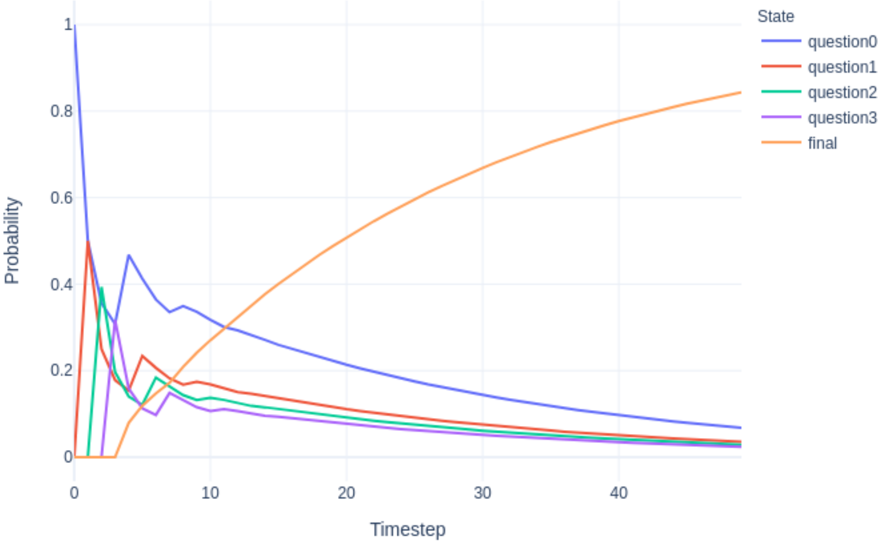


Figure 3: Probability of being at each question over time

$$\frac{1}{\sum_{k=0}^{12} P(\text{harsh}, 2, s_0, s_{2,k})} \sum_{k=0}^{12} T(\text{harsh}, s_{2,k}, s_3) P(\text{harsh}, 2, s_0, s_{2,k}) \approx 0.631.$$

That is, under the greedy strategy, we are less likely to answer question 2 correctly given that we reach it, (of course we are more likely to reach question 2 in the first place). In this section we model our game in the reinforcement learning setting and find the corresponding 'optimal' policy.

Note that as we want to maximize the probability that we get to the final state, it suffices to consider simpler Markov chain where failed guesses go to a separate state, s_{fail}. Apart from this change we keep the same states, and transition probabilities as in the previous section. Let A be the set of actions (guesses), and r, the reward, be 1 if we reach s_4 and 0 otherwise. Define the value of a policy at a state to be

$$V_\varphi(s) = \sum_{s'} T_{\varphi(s')}(s, s')(\mathbb{1}_{s'=s_4} + V(s'))$$

We then find the policy φ_* that maximizes V for all states s using the value iteration algorithm, and set

$$\varphi_*(s) = \text{argmax}_a \{ \sum_{s'} T_a(s, s')(\mathbb{1}_{s'=s_4} + V(s')) \}$$

Table 3 shows the result of running the value iteration algorithm for the lenient variant. In particular, we find that the $\varphi_*(s) = \varphi_{\text{greedy}}$, so the greedy algorithm is in fact optimal.

State	Value	Guess
s_0	0.15	0
$s_{1,0}$	0.20	lower
$s_{1,1}$	0.18	lower
$s_{1,2}$	0.16	lower
$s_{1,3}$	0.15	lower
$s_{1,4}$	0.13	lower
$s_{1,5}$	0.12	lower
$s_{1,6}$	0.11	lower
$s_{1,7}$	0.12	higher
$s_{1,8}$	0.13	higher
$s_{1,9}$	0.15	higher
$s_{1,10}$	0.16	higher
$s_{1,11}$	0.18	higher
$s_{1,12}$	0.20	higher
$s_{2,0}$	0.25	outside
$s_{2,1}$	0.25	outside
$s_{2,2}$	0.23	outside
$s_{2,3}$	0.21	outside
$s_{2,4}$	0.19	outside
$s_{2,5}$	0.17	outside
$s_{2,6}$	0.15	outside
$s_{2,7}$	0.15	inside
$s_{2,8}$	0.17	inside
$s_{2,9}$	0.19	inside
$s_{2,10}$	0.21	inside
$s_{2,11}$	0.23	inside
$s_{2,12}$	0.25	inside
s_3	0.25	0

Table 3: Value and optimal policy at each state

3 Finite deck model

When the deck has a finite number of cards, the game becomes complicated by the fact that probabilities of a certain number or suit is appearing depends on the previous draws. Thus, we ask the question: "how much mileage do we get from memorizing the seen cards?", or similarly: "how closely does the infinite deck model correspond to the actual game?". While it is clear that memory will help the player in the finite model, it is not clear by how much. As cards are drawn at random from the remaining deck, the distribution of values and suits in the decck should be pretty close to uniform so memory may not be that helpful. However, if we arrive at the situation where there is one card in the deck and we are about to answer question 3, the player with memory wins with probability 1 and the player without memory wins with probability $1/m$.

We run computer simulations to compare the perfomance of the φ_{drunk}, φ_{greedy}, and $\varphi_{\text{hyperthymestic}}$ players. Recall that the main difference between the greedy and the hyperthymestic player is that the greedy player doesn't not have any memory of the past cards, and the hyperthymestic player has a perfect memory. Table 4 compares the expected number of drinks for each player under the finite deck model.

We see that in the lenient variant of the game, we are expected have 3.4 fewer drinks if we memorize the cards, and in the harsh variant we are expected to have 4.8 fewer drinks in the standard deck. The ratio between the number of drinks for the greedy player and the hyperthymestic player is around 1.4. Our analysis on the infinite deck model found that N_d grows linearly in m for the greedy player, so we see that the benefit for memorization also increases linearly in m. On the other hand, memory has diminishing returns for large n as the sampling from the deck better approaches the uniform distribution and thus the guess we make using memory is not so different from the guess we make using the greedy strategy.

4 Conclusion

Here we used mathematical and statistical techniques to understand the popular game '*riding the bus*'. Under the idealized infinite model, we fully describe the distribution of the number of drinks for the greedy strategy. Furthermore, we prove that asymptotically, the number of values in a deck does not matter, and the expected number of drinks grows linearly with the number of suits in the deck. We also find that the greedy strategy is in fact optimal under this setting using reinforcement learning techniques.

Additionally, we compute statistics on the the expected number of drinks for different play strategies in the realistic finite deck model and highlight the importance of memory in the game.

variant	strategy	n	m	$E[N_d]$	std(N_d)	median$[N_d]$
harsh	φ_{drunk}	13	4	37.37	38.37	25.00
			10	98.92	101.25	66.00
			20	197.05	197.07	137.00
		20	4	35.28	35.85	24.00
		40	4	32.44	32.84	23.00
	φ_{greedy}	13	4	15.99	16.35	11.00
			10	42.95	42.85	30.00
			20	86.72	85.83	60.00
		20	4	15.27	15.52	11.00
		40	4	14.47	15.16	10.00
	$\varphi_{hyperthymestic}$	13	4	11.11	9.75	9.00
			10	29.60	24.50	25.00
			20	59.59	48.24	50.00
		20	4	11.77	10.42	9.00
		40	4	12.83	12.33	9.00
lenient	φ_{drunk}	13	4	26.05	26.88	18.00
			10	64.07	64.24	44.00
			20	130.58	129.82	91.00
		20	4	27.38	28.07	19.00
		40	4	28.75	29.87	20.00
	φ_{greedy}	13	4	11.85	12.32	8.00
			10	30.32	31.12	21.00
			20	63.30	64.20	43.00
		20	4	12.26	12.63	8.00
		40	4	13.33	13.74	9.00
	$\varphi_{hyperthymestic}$	13	4	8.57	7.50	7.00
			10	22.18	18.27	18.00
			20	44.94	35.61	37.00
		20	4	10.02	9.12	7.00
		40	4	11.66	11.23	8.00

Table 4: Expected number of drinks in the finite model