Simulation of Mia through DEL

Madhur Madhur (s4357469)
Travis Hammond (s2880024)
Emre Erdogan (Guest Student)

04 June 2021

Abstract

Using dynamic epistemic logic and statistical inference to predict the liars and categorise the personalities of players in the dice game Mia.

1 Introduction

Dice and board games are a perfect way to socialise and spend time with friends. Although these games seem very easy to a normal individual, there can be very complex logical calculations that our brains may be doing for us to help us win. One such interesting game is Mia [1] which revolves around bluffing and predicting the other player's likelihood to lie. This game consists of n players and *two* dice and the motive is to keep yourself alive as the rounds progress and to trick other players into believing your lies. This game revolves around the epistemic logic of knowledge and beliefs.

1.1 Game Walk-through

As an explanation of the game, consider 3 players: Alice, Bob and Eve. Alice starts the round by rolling the dice in a dice cup to cover the results. She rolls a 5 and a 4, and the higher roll always counts as the higher denomination, so she publicly (and truthfully) declares to have rolled a 54 and passes the dice, still covered, to Bob. Bob can now decide to believe her and roll the dice without looking, or call her a liar and look at the dice to check. Bob sees no incentive for Eve to have lied on the first round so he rolls the dice. He rolls a 43 and because you can only ever declare a number higher than the one declared previously, he is forced to lie so he declares to have rolled a 55 and passes the dice to Eve. This is a bold move because doubles always count higher so this is the second highest possible roll in the game. Eve considers it unlikely for Bob to have successfully rolled higher than 54, so she calls him a liar, checks the dice and Bob loses a life. Eve can now start a new round. Due to the fact the declared number always has to be higher than before, it becomes less and less likely for someone to have told the truth as the dice are passed around. There is also a special roll called the Mia (1 and 2), that can end the round at any point someone declares it (truthfully or not) because if you are passed the dice after a Mia has been declared you can only either accept losing a life or call the previous player a liar and either lose 2 lives if you are wrong or the previous player loses a life if they lied. A Mia is also the only roll that can be declared after announcing a 66. After losing 6 lives a player dies.

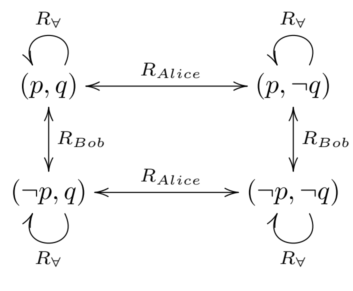


Figure 1: An example Kripke model of two players where p is the fact that Alice is a "liar" player and q is the fact that Bob is a "liar" player.

1.2 Game Simplification

The state space of the dice in this game is comparatively low because we count combinations and not permutations (the die with higher reading is always the tenner position of the rolled value). However, the point of the game at every turn is not to determine the exact value of the dice but to determine whether the player passing the dice is lying and to trick someone into thinking that you are lying even though you are not. We can approximate the former ability (sometimes referred to as the ability to "play the players, not the game") by reducing it to the ability to determine the broad category of lying-friendliness of a given player. To do this we can assign a personality category to each player that determines whether they have a chance of lying in certain situations or not and the goal of the players is now to determine the personality type of each other player. For example, a "naive" type player might only lie when there is no other choice and a "liar" player type might lie even when the odds of her roll are very believable. The complexity of our model can be increased by approximating the lying behavior in more detail and introducing higher-order beliefs and knowledge into the game play. In our simplification of the game the focus is on the modelling of the behaviour of the knowledge that each player have and does not go into the higher degrees of belief.

1.3 Research Question

The research question can then be formulated as: *How can each player learn the lying strategy of the other players through observations in Mia and how will it affect the dynamics of the game?* We model the distributed state of knowledge with a Kripke model where the propositional atoms in each possible world are statements about the personality category each player belongs to. In the beginning of a game, players only know their own personality but as the game progresses the Kripke model (for an example, see Figure 1) should simplify until each player infers the personality of the preceding player.

Additionally, we wanted to use second-order knowledge and beliefs by extending our model to include statements such as "Alice believes that Bob knows that Alice is a liar". This addition combined with the player's ability to switch the personality categories if such a fact is known, could have also be an interesting dynamic to investigate. Unfortunately, we did not have time to analyze these dynamics.

Algorithm 1: Dice Announcement Algorithm (lying probability = $i \in \{0.2, 0.5, 0.8\}$)

```
Data: Previous/Initial Dice Roll
Result: Current Roll
Initialization;
Throw dice and get the current roll;
Get a random number r where 0 ≤ r ≤ 1;
if First turn of the round or current roll > previous roll then
    if r ≥ i then
        Say the current roll (e.g. truth);
    else
        Say a roll > previous roll with probability p where p = 1/m and m is the number of rolls
          that can be said in the current turn (e.g. unnecessary lie);
    end
else
    Say a roll > previous roll with probability p where p = 1/m and m is the number of rolls that
      can be said in the current turn (e.g. necessary lie);
end
```

2 Agent Model

Our agent modelling is based on our answers to the next four questions:

1. When does an agent lie/not lie?
2. What does an agent say when lying?
3. When does an agent call out the preceding agent?

4. How does an agent classify the preceding agent?

Algorithm 2: Call-Out Algorithm

Data: Dice Roll
if *Player's turn* **then**
  Gather evidence from previous public lies and truths;
  Call-out the previous player if evidence points towards a different world than the one announced;
else
  Observe public truths and public unnecessary lies of each player;
  Update world view based on the observed round;
end

Our answers are straightforward. For the first one, we use fixed lying probabilities of 0.2, 0.5, and 0.8: An agent always lies with the probability assigned to it. These probabilities works are the three unique world view that each player has. For the second question, we use a uniform distribution for the lying probabilities so that if an agent decides to lie, it can say any dice value that can be said at that time with the same probability. For the third one, we use a threshold value: An agent calls-out the preceding agent with 0.5 probability at default. For the last one, we count the number of unnecessary lies and truths so that an agent can infer the lying probabilities assigned to the other agents and its call-out threshold value changes with these inferences. For the announcement and call-out algorithms, see Algorithms 1 and 2.

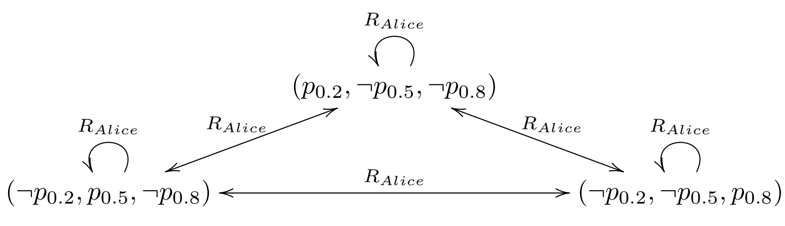
2.1 Kripke Model

In the game, all the announcements, the call-outs, the results of call-outs, and the fact that players can only have either 0.2, 0.5 or 0.8 lying probability are common knowledge. This allows agents to acquire more information about others as they observe their actions (e.g. call-outs, lies etc.) as the game continues and hence, the inferences about lying probabilities become more precise. For an example of how an agent's perspective of another agent is at the beginning of a game and how it evolves and is simplified over time, see Figure 2 (the overall perspective requires 3^n states which we cannot show here).

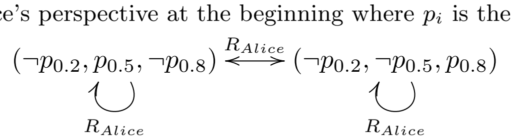
3 Experiment Setup

3.1 Gameplay

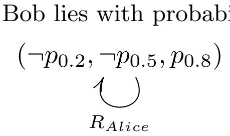
In our experiment, six agents named Joey, Phoebe, Ross, Rachel, Monica, and Chandler play the game (in this order) and the assigned lying probabilities are 0.8, 0.5, 0.2, 0.8, 0.5, and 0.2, respectively. An agent starts with six lives and is eliminated from the game when it loses all of its lives. The last agent that has at least one life wins the game.



(a) Alice's perspective at the beginning where p_i is the fact that Bob lies with probability $i \in \{0.2, 0.5, 0.8\}$.



(b) Alice's perspective after n rounds where Alice's updated inference is that Bob lies with 0.65 probability ($0.5 < 0.65 < 0.8$).



(c) Alice's perspective after $m > n$ rounds where Alice's updated inference is that Bob lies with 0.78 probability ($0.78 \approx 0.8$).

Figure 2: Alice's evolving perspective of Bob

3.2 Implementation

The implementation is done in JavaScript, and a webpage is active at https://dashdeckers.github.io/Mia/. The implementation emulates the agent model from section 2. Below are the instructions on how to simulate the game:

1. **Setup**: Refreshes the gameplay to default with six players and resets the game state.
2. **Step**: Simulates the game for one turn, essentially moves forward the game in single-step fashion.
3. **Play 10 games**: Simulates 10 games.
4. **Play 100 games**: Simulates 100 games.
5. **Toggle AI (true/false)**: Enables/Disables the AI mechanism designed for inferring the lying probabilities of agents.

As the simulation progresses, one can find the agents' current perspectives in the left side of the webpage that contains the following information:

1. **"lying_probability"**: Agent's lying probability (private). It is set at the beginning and does not change throughout the simulation.
2. **"game_wins"**: Shows how many games the agent has won so far (public).
3. **"public_lies"**: Shows how many necessary lies of the agent are public.
4. **"public_truths"**: Shows how many truths of the agent are public.
5. **"inference(agent_i)"**: Agent's current inference on agent i's lying probability (private).
6. **"truth(agent_i)"**: Agent i's actual lying probability (private).

On the right side of the webpage, one can find the running narrative of the game. Figure 3 shows how the simulation looks like after a player wins one game.

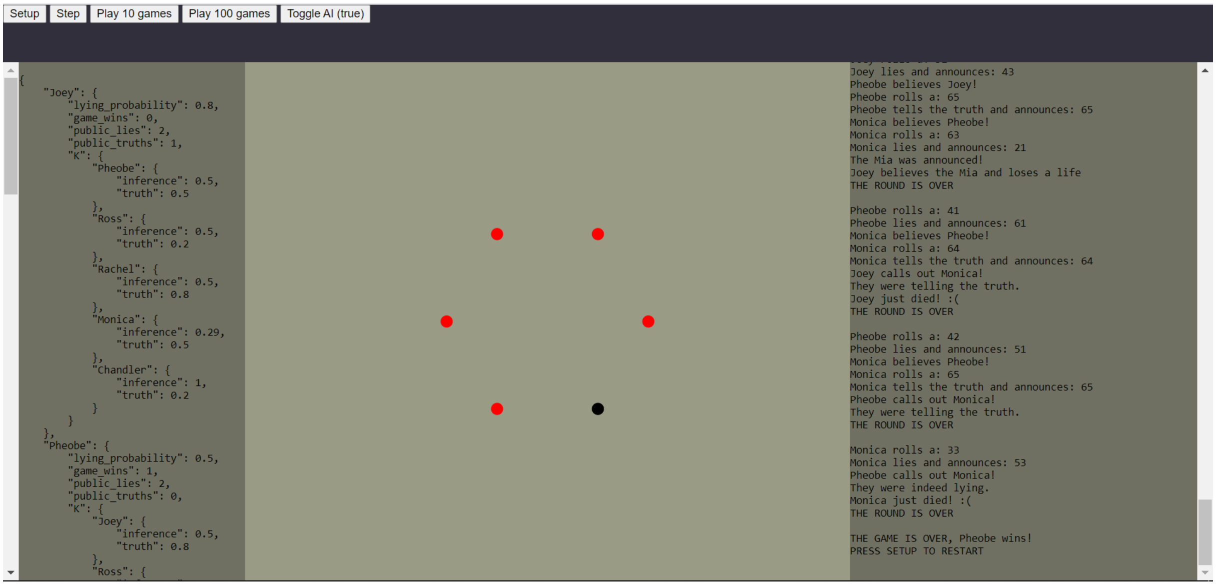
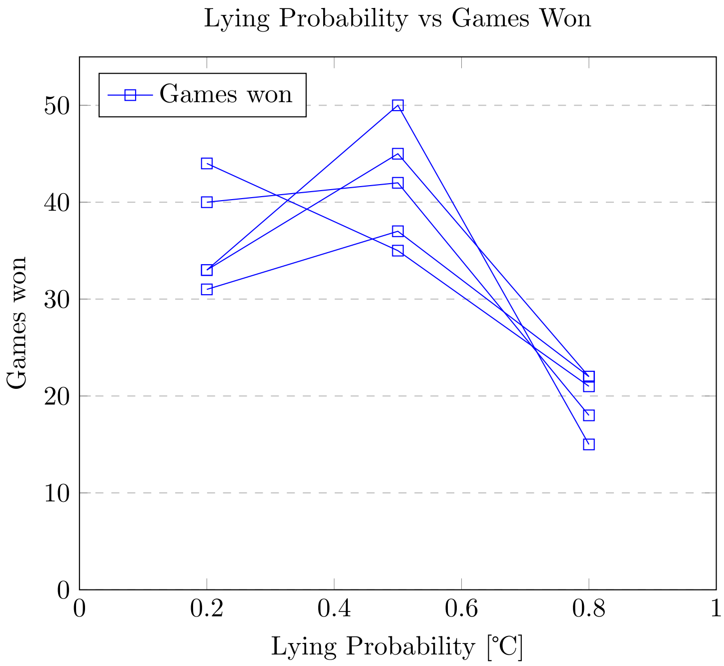


Figure 3: Simulation of Mia

4 Results

We saw that the nature of each player influence how they are treated by other player agents in the game, which in turn determines their changes of winning the game. In figure 4 it can be seen that as a player with a high lying tendency end up being calling out more and loosing game. This observation also point out that the agents with less lying tendency end up covering up their lies better in the game.



5 Conclusion

6 Discussion

References

[1] Wikipedia contributors. Mia (game) — Wikipedia, the free encyclopedia, 2021. [Online; accessed 17-June-2021].