

https://dspace.org/

Estimating skills in discrete pursuit-evasion games

Gomes, Byron John

2023
University of the Witwatersrand, Johannesburg
https://hdl.handle.net/10539/38475

Downloaded from a DSpace repository. https://dspace.org/

Estimating skills in discrete pursuit-evasion games

Byron John Gomes



A research report submitted the Faculty of Science, University of the Witwatersrand, Johannesburg, in partial fulfilment of the requirements for the degree of Master of Science.

November 16, 2023

Declaration

I declare that this research report is my own, unaided work. It is being submitted for the Degree of Master of Science at the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination at any other University.



(Signature of candidate)

16 day of November 20 23 at Midrand

Abstract

Game Theory is a well-established field in mathematics, economics, and computer science, with a rich history of studying n-person, zero-sum games. Researchers have utilized the best computational power of their time to create computational players that are able to beat the best human players at complex two-player, zero-sum games such as Chess and Go. In the field of Reinforcement Learning and Robotics, these types of games are considered useful environments to conduct experiments about agent behavior and learning.

In this research report we explore a subset of discrete skill-dependent pursuit-evasion games upon which we build a framework to estimate player skills. In this game environment a player's skill determines the actions available to them in each state and the transition dynamics resulting from the chosen action. The game offers a simplified representation of more complex games which often have vast state and action spaces, making it difficult to model and analyze player behavior.

In this game environment we find that players with incorrect assumptions about an opponent's skill perform sub-optimally at winning games. Given that knowledge of an opponent's skill impacts on player performance, we demonstrate that players can use Bayesian inference to estimate their opponent's skill, based on the action outcomes of an opponent. We also demonstrate that skill estimation is a valuable exercise for players to undertake and show that the performance of players that estimate their opponent's skill converges to the performance of players given perfect knowledge of their opponent's skill. This research contributes to our understanding of Bayesian skill estimation in skill-dependent pursuit-evasion games which may be useful in the fields of Multi-agent Reinforcement Learning and Robotics.

Acknowledgements

Firstly, I would like to acknowledge the financial support provided to me by the Standard Bank of South Africa in pursuit of this degree.

Secondly, I would like to acknowledge my supervisors Benjamin Rosman and Dylan Shell and thank them for having seemingly bottomless patience, endless wisdom, and generally good senses of humour. I am truly grateful to have worked with such incredible academics who have not only bestowed on me some valuable academic insights and knowledge but also valuable life lessons.

Lastly, I would like to acknowledge my wife Neesha Fakir whose immense support and encouragement saw me through many a moment of discouragement and despair. Her accomplishments inspired me to pursue this degree and I believe that I am a better person for it.

Contents

1 Introduction **1**

2 Background **4**
2.1 Pursuit-Evasion Games . . . 5
2.2 Markov Games . . . 5
2.3 Minimax and Expectiminimax . . . 6
2.4 Bayesian Inference . . . 8

3 Related Work **10**
3.1 Multi-agent Reinforcement Learning . . . 10
3.2 Bayesian Execution Skill Estimation . . . 10
3.3 Opponent Modelling in Adversarial Domains . . . 11
3.4 Extended Form Games and Sequential Equilibria . . . 11
3.4.1 The Chain Store Paradox and Irrational Players . . . 12
3.4.2 Trembling Hand Perfect Equilibria . . . 13
3.4.3 Sequential Equilibria . . . 14

4 Research Objectives and Methodology **16**
4.1 Markov Game Environment . . . 17
4.1.1 Transition Dynamics . . . 18
4.1.2 Win Conditions and Scoring . . . 19
4.1.3 Game Instances . . . 20
4.2 Game Evaluation . . . 21
4.3 Skill Estimation . . . 23

5 The Expected Reward Function **26**
5.1 Computing Expected Rewards . . . 27
5.1.1 Convergence of Empirical Results to Closed-Form Results . . . 27
5.1.2 Expected Reward Versus Skill . . . 28
5.1.3 Expected Reward Versus Assumptions . . . 31
5.2 Effects of the Failure Factor . . . 34
5.3 Effects of Discounting the Expected Reward . . . 38
5.4 Effects of a Different Reward for Ties . . . 41

6 Skill Estimation **47**
6.1 Skill Estimation of Expectiminimax Players . . . 48
6.2 Estimating Skills with Increased Failure Factor . . . 53
6.2.1 The Convergence of Skill Estimates and Reaching Equilibrium . . . 56
6.3 The Validity of Assuming Rationality When Estimating Skill . . . 59

7 Conclusion **66**
7.1 The Expected Reward Function 66
7.2 Skill Estimation 67
7.3 Future Work 68

A Skill Estimation Graphs **69**
A.1 Skill Estimation of Expectiminimax Players Continued 69
A.2 Estimating Skills with Increased Failure Factor Continued 74
A.3 Estimating the Skill of Random Players Continued 79

Bibliography **84**

Chapter 1

Introduction

There is a rich history in mathematics, economics, and computer science of studying n-person, zero-sum games. John von Neumann [1], John Nash [2] and many others had set the mathematical foundations of Game Theory by solving examples of these types of games. Utilising the best available computational power of their day, researchers have been able to create computational agents that can beat humans at more complex adversarial, two-player, zero-sum games such as Chess with IBM's Deep Blue algorithm [3] and Go with Google's AlphaGo algorithm [4]. In the field of Reinforcement Learning (RL) and Robotics these types of games are often considered useful environments in which to conduct experiments about agent behaviour and learning.

Continuing in the vein of solving games such as Chess and Go, we are interested in exploring the world of adversarial, two-player, zero-sum games. As a starting point, it is often assumed that the players act rationally when playing such games [5], i.e. each player strives to maximize wins/gains and minimize losses. It is also common to assume in this setting that there are a finite number of states of the game and each player has a finite number of actions to take in each state of the game. In such settings, players are typically given a reward for winning the game. We assume the reward given to players for each game is fixed and predetermined. We also assume the transition dynamics for each player between states is fully understood by the players and is predefined to be either stochastic or non-stochastic.

In the literature we find examples of skill estimation [6][7], or opponent modelling [8][9], but research on integrating skill estimation into opponent modelling is still "under-explored" [9]. The world of adversarial, two-player, zero-sum games provides us with a setting in which we can explore opponent modelling and skill estimation. By doing this we can test hypotheses about players' performance in these games when players are required to guess or estimate the skill of an opponent.

The types of games we wish to explore also rely on the skill of players. Each player's skill determines the actions available to them in each state and determines the transition dynamics resulting from the chosen action in a given state. The skill of each player is only known to that player and neither player has knowledge of their opponent's skill. This allows us to explore the impact of the assumptions each player has about their opponent's skill on their own play and the outcome of the game.

Specifically we are interested in the world of adversarial, two-player, zero-sum, finite-horizon, pursuit-evasion games. We want the nature of our game to be adversarial as this provides players with an incentive to estimate skill in order to optimise their performance relative to their opponent. Having a game that is two-player helps to simplify the reasoning that players need to use when optimising their strategies. This allows us to build a tractable algorithm for the game.

Choosing the games to be zero-sum makes the trade-off of rewards between players clear. If one player performs better by estimating the skill of its opponent, the opponent necessarily performs worse as a result. Giving the game a finite horizon also assists with the tractability of the solution and prevents the games getting stuck in infinite loops. Finally, we choose pursuit-evasion games as these are compatible with all of the previous choices of game type and are widely studied in the literature [10].

In this research report we introduce a cat and rat pursuit-evasion game in which a cat player chases a rat player around a directed, weighted graph which may either be randomly generated or constructed intentionally to illicit specific player behaviour.

In this setting we test the following hypotheses about players competing in these types of games:

1. Incorrect assumptions cause harm: When players make incorrect assumptions about their opponent's skill it can only harm their performance. In Chapter 5 we present experimental evidence that players do not attain their optimal performance when their assumption about their opponent's skill does not match the true skill of the opponent.

2. Players can estimate skills from observed play: The outcomes of actions made by opponents can provide information to players about their opponent's skill and allow for skill estimation. In Chapter 6 we demonstrate experimentally that players can use Bayesian inference to localise their opponent's skill based on observed play.

3. Skill estimation is valuable: Players that estimate the skill of their opponent perform better than players that do not estimate skill. In Chapter 6 we provide experimental evidence that the performance of players using Bayesian inference to estimate their opponent's skill converges to the optimal performance of players given perfect knowledge of the opponent's skill.

In Chapter 2 we cover some background knowledge required before we introduce our research objectives and framework. Chapter 3 considers related work that provides context for the problem we have chosen to investigate and supports our reasons for researching this topic.

In Chapter 4 we define our research hypotheses and outline how they are tested. This includes detail on the game environment to be used in our experiments. A notion of player skills is embedded into the game environment in such a way that the transition dynamics are dependent on skill. We also design an algorithm that can be used to compute expected reward as a function of player skills and players' assumptions about their opponent's skill in this game environment.

Properties of this expected reward function are explored in Chapter 5 and we provide evidence in support of our hypothesis that bad assumptions cause harm. This is the first major contribution of this research. We also explore the effects of the different parameters introduced into our game environment on the expected reward function.

Finally, in Chapter 6 we estimate a player's skill based on observed play in our game environment. We provide evidence to support our hypothesis that players can estimate skills from observed play. This is the second major contribution of this research.

This skill estimate is then used by players in Chapter 6 to update the their policy and improve their performance in the face of uncertainty about the opponent's skill. We provide evidence to support our hypothesis that skill estimation is valuable. This is the third major contribution of this research.

Finally in Chapter 7 we draw conclusions from our experiments and comment on the results of our hypothesis testing. We note the effects of different game parameters on the results of the games

for the players. We also comment on the ability to estimate skills in the proposed game environment using the proposed algorithm. Finally, we comment on possible future work that could answer further questions about estimating player skills.

Chapter 2

Background

In the previous chapter, we briefly discussed the idea of adversarial, two-player, zero-sum, finite-horizon, pursuit-evasion games. First we define each of these descriptors for the games we are interested in.

- Adversarial: games in which two or more players have opposing goals, and each player aims to prevent their opponent(s) from achieving their objectives. In adversarial games, the players are in competition with one another and attempt to outmaneuver or outsmart their opponents to achieve their goals.
- Two-player: games that involve exactly two players. In two-player games, each player has only one opponent to compete against, which makes the game simpler and more manageable than games with more players.
- Zero-sum: games where the total gains and losses of the players are equal and opposite. This means that any gain made by one player must come at the expense of the other player. The term "zero-sum" reflects the fact that the sum of the players' gains and losses is always zero.
- Finite-horizon: games in which there is a fixed number of rounds or moves that the players can make. In other words, the game has a predetermined length or duration, and each player must make the most of their opportunities within that time frame. This is in contrast to infinite-horizon games, where there is no fixed endpoint and the game can continue indefinitely.
- Pursuit-evasion: games in which one player (the pursuer) tries to capture or tag another player (the evader) by pursuing them through a space. The evader's goal is to avoid capture, while the pursuer's goal is to catch the evader. Pursuit-evasion games can take place in a physical space (such as a tag game on a playground) or a virtual space (such as a video game).

In this chapter we expand on some of these ideas and recall some foundational concepts. We begin by discussing pursuit-evasion games in Section 2.1. We then define Markov Games as a model for these types of games in Section 2.2. We then consider Minimax and Expectiminimax as tools in this setting to determine the expected reward for a player in Section 2.3. And finally we consider Bayesian Inference as a method for updating the probability of a hypothesis based on observed play in Section 2.4.

2.1 Pursuit-Evasion Games

We have selected pursuit-evasion games as the type of game we wish to base our game environment on. Shen et. al. [11] have made use of discrete-time pursuit-evasion games to answer questions about theoretical controls for ground-based robots. Similarly, Wan et. al. [10] have also made use of discrete-time pursuit-evasion games to improve approaches to decision-making in the field of multi-agent reinforcement learning. Chung, Cohen, and Graham [12] explored the game theoretic properties of pursuit-evasion games on graphs and found some counter-intuitive behaviour can occur in these types of games for players following optimal mixed strategies.

The literature suggests that pursuit-evasion games are widely studied [10] and that the discrete-time variety of these games are particularly popular. In this research report we choose to make use of a subset of discrete-time pursuit-evasion games that use a network graph as the state space. This subset of games allows us to build a framework in which we can test our hypotheses and provide us with a sufficient number of free parameters to explore different aspects of player behaviour and decision-making in the game. This subset of games also allows us to construct a tractable, closed-form solution for the game upon which players can base their strategies. Pursuit-evasion games are also compatible with the other features we desire: adversarial, two-player, zero-sum, and with finite-horizon.

2.2 Markov Games

The game environment we wish to explore relies on decision making where the outcome is partly random, and partly determined by the actions of the player. A Markov game, as introduced by Shapley [13], is a type of repeated game played by one or more players in game theory. It involves probabilistic transitions and is played in stages where the game starts in a particular state. Each player selects an action, and the payoff for each player depends on the current state and the chosen actions. The game then moves to a new state randomly, with the distribution of the new state being dependent on the previous state and the actions taken by the players.

The Markov Game setting presented by Bowling and Veloso [14], and Chang [15] provide us with a mathematical framework for modelling this type of problem. Consider one of the two players in an adversarial, two-player, zero-sum, finite-horizon, pursuit-evasion Markov game with a finite number, T, of time steps or moves. Without loss of generality, we can refer to the two players as the rat and the cat. For a player playing as the rat, and their opponent playing as the cat we can represent our Markov game as a tuple $(S, A^s_{\text{rat}}, A^s_{\text{cat}}, P^a_{\text{rat}}, P^a_{\text{cat}}, R^a_{\text{rat}}, R^a_{\text{cat}}, \gamma)$ where:

- S is the finite, non-empty set of all states of the Markov game and $s \in S$ is a particular state of the game;
- A is the finite, non-empty set of all actions available to players in the Markov game;
- $A^s_i \in A$ is the finite, non-empty set of all possible actions available to player $i \in \{\text{rat}, \text{cat}\}$ in state s and $a \in A^s_i$ is a particular action that may be taken by player i when in state s;
- $P^a_i(s, s')$ is the probability that player i successfully transitions from state s to state s' when performing action a;
- $R^a_i(s, s') \in \mathbb{R}$ is the reward received by player i after transitioning from state s to state s' due to action a;

- γ is the one-period discount factor players use to discount their rewards.

This looks very similar to the Markov Decision Process (MDP) framework described by Bellman [16] and later by Wrobel [17], except that we have two players instead of one. Markov games are a natural extension of MDPs to multiple agents. In the event that $P_i^a(s, s') = 1$ for all players, states, and actions, we would have a game where all transitions occur with absolute certainty and could be solved with a Minimax algorithm. Where $0 \leq P_i^a(s, s') < 1$ for at least some players, states, and actions, we have a game where transitions do not always occur with absolute certainty. This type of game would need to be solved by an algorithm such as Expectiminimax.

At some point in time $t = \{0, 1, 2, \ldots, T\}$, player i finds itself in state $s_i(t) \in S$. Let $a_i(s_i(t)) \in A_i^s$ be a playable action available to player i at time t when in state $s_i(t)$. We also have the transition dynamics given by probability $P_i^a(s, s') = Pr(s_{t+1} = s' | s_t = s, a_t = a)$.

Let Π_i be the set of all possible action policies for i. The policy $\pi_i \in \Pi_i$ contains a set of ordered actions $\{a_i(s_i(0)), a_i(s_i(1)), ..., a_i(s_i(T))\}$ that player i takes at each time $t = \{0, 1, 2, \ldots, T\}$ from each state $\{s_i(0), s_i(1), ..., s_i(T)\}$.

Now we introduce the concept of an optimal action $a_i^*(s_t)$ which maximises the reward $R_i^a(s, s')$ received by player i at time t. With this we can now consider the concept of the optimal policy $\pi_i^* \in \Pi_i$ which maximises the rewards R received by player i over the entire game. That is π_i^* is the policy containing the set of actions, $\{a_i^*(s_i(0)), a_i^*(s_i(1)), ..., a_i^*(s_i(T))\}$, that maximise the expected discounted reward for player i.

2.3 Minimax and Expectiminimax

When playing Markov games we can use the Minimax algorithm, discussed by Von Neumann and Morgenstern [1], to solve the game when the transition dynamics are non-stochastic. The Minimax algorithm is used to determine the best move for a player, assuming that the opponent is also playing optimally. The algorithm works by evaluating all possible moves and their resulting outcomes, assigning a score to each outcome based on its desirability, and then choosing the move that maximizes the player's score while minimizing the opponent's score.

Expectiminimax is a variant of the Minimax algorithm proposed by Michie [18] that is used in decision-making under uncertainty in two-player games. The Expectiminimax algorithm considers the expected reward of each possible move, rather than just the worst-case (minimum) or best-case (maximum) outcomes, as the Minimax algorithm does. It does this by taking into account the probabilities of different outcomes occurring, rather than just considering the minimum and maximum values. This allows the Expectiminimax algorithm to be more robust in the face of uncertainty, as it can better handle situations where the outcome of a move is not certain.

While the reward function discussed in Section 2.2 gives us an instantaneous reward after successfully transitioning from one state to another, the Minimax and Expectiminimax algorithms give us the sum of expected future rewards when each player behaves optimally. This allows players to differentiate between potential winning and losing actions, particularly when rewards in the game are received infrequently or only at the end of the game.

Let $V_i(s, t, \gamma)$ be the value of the game player i when playing the game from state s at time t with discount factor γ. This value is determined by calculating the expected reward of the game to player i when in state s at time t. Let s' be a child state of s' where the probability of player i reaching state

s' from state s is given by $P_i(s, s')$.

Algorithm 1 Calculate the expected reward $V_i(s, t, \gamma)$

```
1: function V_i(s, t, γ):
2: if state is terminal state or t = T then
3:    return  terminal reward R_i
4: end if
5: if opponent is to play in current state s then
6:    // Return reward of minimum-valued child state s'
7:    let V_i(s, t, γ) := +∞
8:    for all children s' of current state s do
9:       V_i(s, t, γ) := min(V_i(s, t, γ), γ × V_i(s', t + 1, γ))
10:   end for
11: else
12:   if player i is to play at current state s then
13:      // Return reward of maximum-valued child state s'
14:      let V_i(s, t, γ) := −∞
15:      for all children of current state do
16:         V_i(s, t, γ) := max(V_i(s, t, γ), γ × V_i(s', t + 1, γ))
17:      end for
18:   else
19:      // Return expected reward of all child states' values
20:      let V_i(s, t, γ) := 0
21:      for all children s' of current state s do
22:         V_i(s, t, γ) := V_i(s, t, γ) + γ × [(P_i(s, s') × V_i(s', t + 1, γ)]
23:      end for
24:   end if
25: end if
```

Using the notation already established in this chapter, we present the pseudocode for the Expectiminimax algorithm in Algorithm 1. With this algorithm we can assess the discounted expected reward of the game with respect to each player for discount factor $\gamma \leq 1$.

This recursive Expectiminimax algorithm finds the optimal move in a two-player game when the other player is also making choices that can affect the outcome. In lines 2–4 of Algorithm 1 we determine if we are either at a terminal state (the goal) or terminal time T, and if so, we return the reward for each player (e.g. +1 for a win, and -1 for a loss). Otherwise, we consider each player during their turn. If it is the opponent's turn to play, looking at lines 5–10 of Algorithm 1, we compute the minimum-valued child state that results from opponent's actions. In lines 12–17, we similarly compute the maximum-valued child state that results from player i's actions. Finally in lines 19–23 we compute the expected reward for all child states' values.

Using Algorithm 1 we can construct an expected reward or value matrix $V_i(s, t, \gamma)$ for each state s at each time t. The rows of this matrix represent the time step in the game and the columns represent each state. The last row of each player matrix is populated with the rewards that that player receives at time T in each state. Using our assumption that each player i is the maximising their reward,

and that the rewards are zero-sum, we can recursively solve for the discounted expected rewards and consequently each players' optimal policy (π_i^*) at the previous time step.

Consider that the rat and cat only have a estimates or assumptions about what their opponent's optimal policy is. We define the rat's and cat's estimates or assumptions of the opponent's optimal policy as $\hat{\pi}^*_{\text{cat}}$ and $\hat{\pi}^*_{\text{rat}}$ respectively. We also define the concept of the player transition probabilities under their respective optimal policies as $P^{\pi^*_{\text{rat}}}_{\text{rat}}(s, s', \gamma)$ and $P^{\pi^*_{\text{cat}}}_{\text{cat}}(s, s', \gamma)$. Each player's rewards under their respective optimal policies $R^{\pi^*_{\text{rat}}}_{\text{rat}}(s, s')$ and $R^{\pi^*_{\text{cat}}}_{\text{cat}}(s, s')$. Furthermore, we define player cat's transition probabilities and rewards under player rat's estimate or assumption of the cat's optimal policy, as $P^{\hat{\pi}^*_{\text{cat}}}_{\text{cat}}(s, s', \gamma)$ and $R^{\hat{\pi}^*_{\text{cat}}}_{\text{cat}}(s, s')$ respectively.

We now have the following equations that define the expected reward function of the rat, its optimal policy, and the rat's estimate or assumption of the cat's optimal policy:

$$V_{\text{rat}}(s, t-1, \gamma) = \sum_{s'} P^{\pi^*_{\text{rat}}}_{\text{rat}}(s, s', \gamma)[R^{\pi^*_{\text{rat}}}_{\text{rat}}(s, s') + \gamma V_{\text{rat}}(s', t, \gamma)], \tag{2.1}$$

$$\pi^*_{\text{rat}}(s, t-1, \gamma) = \underset{a^s_{\text{rat}} \in A^s_{\text{rat}}}{\text{argmax}} \{P^{a^s_{\text{rat}}}_{\text{rat}}(s, s', \gamma)[R^{a^s_{\text{rat}}}_{\text{rat}}(s, s') + \gamma V_{\text{rat}}(s', t, \gamma)]\}, \tag{2.2}$$

$$V_{\text{rat}}(s, t-2, \gamma) = \sum_{s'} P^{\hat{\pi}^*_{\text{cat}}}_{\text{cat}}(s, s', \gamma)[R^{\hat{\pi}^*_{\text{cat}}}_{\text{cat}}(s, s') + \gamma V_{\text{rat}}(s', t, \gamma)], \tag{2.3}$$

$$\hat{\pi}^*_{\text{cat}}(s, t-2, \gamma) = \underset{a^s_{\text{cat}} \in A^s_{\text{cat}}}{\text{argmin}} \{P^{a^s_{\text{cat}}}_{\text{cat}}(s, s', \gamma)[R^{a^s_{\text{cat}}}_{\text{cat}}(s, s') + \gamma V_{\text{rat}}(s', t, \gamma)]\}. \tag{2.4}$$

Note the above equations are computing the game value from the rat's perspective using the rat's estimate or assumption about the cat's optimal policy do so. The cat would have analogous equations to compute the game value from its perspective using its estimate or assumption about the rat's optimal policy.

For the majority of our experiments we use $\gamma = 1$ in our computation of the Expectiminimax algorithm (i.e. no discounting). We consider the effects of discounting (i.e. $\gamma < 1$) in some of our experiments to determine the impact of discounting the expected reward on player performance and optimal policies.

2.4 Bayesian Inference

Now that we have a way to model our game, a method for evaluating the expected reward of different game positions, and an algorithm to determine optimal policies for players, it is useful to consider how we are going to estimate an opponent's skill. An extremely powerful tool in solving this particular problem is Bayesian Inference [19]. We start with specifying Bayes' Theorem:

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}, \tag{2.5}$$

where:

- H is the hypothesis.
- $P(H)$ is the prior probability distribution of the hypothesis before accounting for observed data.
- E is the evidence or observed data.

- $P(H|E)$ is the posterior probability distribution of the hypothesis after accounting for observed data.
- $P(E|H)$ is known as the likelihood and is the probability of observing the evidence given the hypothesis.
- $P(E)$ is known as the marginal likelihood. Often this part is ignored when comparing different hypotheses as it remains constant regardless of the hypothesis under investigation.

The basic intuition behind Bayes Theorem is that it allows us to compute the probability of a hypothesis given some observed evidence or data, by reversing the order of the conditional probabilities. In other words, it enables us to estimate the probability of a hypothesis being true, given the observed evidence or data, and the prior probability of the hypothesis being true.

In our game environment, players use Bayes Theorem to update their hypothesis about the opponent's skill based on the evidence of observed play and a prior probability distribution of the opponent's skill.

Chapter 3

Related Work

While there is a rich history in the study of adversarial, two-player, zero-sum games, in this chapter we concern ourselves with recent developments in games involving skill and skill estimation specifically. Here we outline related work in this field.

3.1 Multi-agent Reinforcement Learning

In his PhD thesis, Hernandez [8] introduces the concept of a Multiagent Reinforcement Learning (MARL) stack which enables agents equipped with learning algorithms to train via simulated learning in a MARL pipeline environment. Hernandez specifically explores the context of two-player, zero-sum games and the different approaches to granting opponent awareness. Hernandez also demonstrates through empirical results that the learning dynamics of agents are greatly affected by the sequence of opponents faced during training.

In this research, Hernandez considers opponent modelling in two-player, zero-sum games, but does not consider introducing the concept of skill into the game. Introducing skill into the game environment requires players to perform additional reasoning about the game and how skills affect the outcomes. This necessitates the players' need to be aware of their opponent's skill too.

3.2 Bayesian Execution Skill Estimation

Archibald et. al. have previously investigated how the notions strategic and execution skill interact in the domain of computational pool/billiards [6]. Archibald and Nieves-Rivera [7] have built on this previous work to address the problem of estimating the execution skill of an agent when acting in domains with continuous action spaces that depend on selecting good actions and executing those actions accurately.

In this latest paper the authors present a Bayesian framework that allows agents to reason about action observations. They are able to demonstrate in a toy domain and in the domain of computational billiards that this framework outperforms previous methods under identical conditions.

In this research, Archibald and Nieves-Rivera model the skill of the player as it pertains to the outcomes of shots taken by the player in computational pool. However, they do not consider any modelling of the opponent since, in this environment, the player wins 70% of the time regardless of who the opponent is. In the other 30% of games, defensive strategies were employed. So while skill is

embedded in the game of computational pool, there is little value in estimating the opponent's skill or modelling the opponent at all.

3.3 Opponent Modelling in Adversarial Domains

Skill estimation is a specific problem considered in the broader problem of opponent modelling. Nashed and Zilberstein [9] survey a variety of approaches to modelling the ability of opponents in adversarial domains. In this paper the authors propose a novel framework that allows comparison between the different modelling methods. The opponent modelling and skill estimation problems are typically considered separately in the approaches surveyed.

In their discussion on open problems and future research directions they note the modelling of failed actions as something that is not currently captured in the assessed opponent modelling approaches. They propose that data about failed actions can be used to improve skill estimates when modelling the capabilities of opponents. The authors also point out that, "[skill estimation] is still an under-explored area of research and has yet to be integrated in opponent modeling systems at any notable scale." [9]

3.4 Extended Form Games and Sequential Equilibria

In the realm of game theory, Nash equilibria [2] stand as pivotal concepts that illuminate the intricate interplay between strategic decision-making and optimal outcomes. Named after the renowned mathematician and Nobel laureate John Nash, these equilibria represent situations in which each participant's choice of strategy remains indifferent to alternate strategies with the same payoff.

Consider a two player game where player 1 has strategies $s_1 \in S_1$ and player 2 has strategies $s_2 \in S_2$. The payoff of this game to player i is given by $\text{payoff}_i(s_1, s_2)$. "A strategy s_1^* is a Best Response by player 1 to a strategy s_2 for player 2 if $\text{payoff}_i(s_1^*, s_2) \geq \text{payoff}_i(s_1, s_2)$ for all strategies $s_1 \in S_1$. A pair of strategies (s_1^*, s_2^*) is in Nash Equilibrium if s_1^* is a Best Response by player 1 to s_2^*, and s_2^* is a Best Response by player 2 to s_1^*."[20]

In his chapter on Strategic Equilibrium Van Damme [21] reflects on the work of Selten [22], where Selten pointed out that "in extensive form games, not every Nash equilibrium can be considered self-reinforcing". A self-enforcing agreement or equilibrium occurs when neither player would deviate from their optimal strategy or policy π_i^* because doing so can only result in a less desirable outcome for the deviating player. Van Damme continues to note that two requirements need to be met in order for players to reach a self-enforcing agreement:

- no player can deviate profitably from their optimal strategy unless another player deviates from their optimal strategy,
- the expectation that no player deviates from their optimal strategy is rational.

Consider the following simple game where player 1 can choose between actions A and B, and player 2 can choose between actions C and D. The payoff matrix of this game is as follows:

		Player 1	
		A	B
Player 2	C	5	3
	D	3	4

In this game, if player 1 chooses action A, and player 2 chooses action C, then the payoff of the game for both players is 5. If either player deviates from these strategies, then the other player can deviate profitably by changing their strategy too. For example, if player 1 deviates from the strategy of choosing A, and instead chooses B, then if player too sticks to their original strategy, the payoff will be 3. However, if player 1 chooses B, then it would be rational for player 2 to also deviate from choosing C and instead choose D which now results in a payoff of 4. Thus both requirements provided by Van Damme are satisfied and the action pair (A,C) is an example of a self-enforcing agreement.

We note that the action pair (B,D) is also a Nash Equilibrium since action D is player 2's best response to action B and action B is player 1's best response to action D. However, it is not a self-enforcing agreement since both players can deviate, and reasonably expect the other player to profitably deviate resulting in a higher payoff for both.

In this section we are going to discuss various types of equilibria that can be defined and we will build up to a definition of sequential equilibria, but first we need to introduce some notation related to extensive form games.

A game g is a finite extensive form game with perfect recall if it has the following properties:

- a collection of I players,
- a game tree K specifying the physical order of play,
- for each player i a collection H_i of information sets specifying the information available to a player when it is their turn to move. Note that H_i is a partition of the set of decision points of player i, and if two nodes s and s' are in the same element $h \in H_i$ then i cannot distinguish between s and s',
- for each information set h there is a feasible set of choices or actions A_h,
- the probabilities associated with chance moves are specified,
- for each end point z of the tree and each player i there is a payoff or reward of $V_i(z)$ for player i reaching z.

In games with perfect information one may require as a condition for π_i^* to be self-enforcing, that it satisfies:

$$V_{i,h}(\pi_i^*) \geq V_{i,h}(\pi_i^* \backslash \pi_i) \text{ for all } i, \text{ all } \pi_i \in \Pi_i, \text{ all } h \in H_i \tag{3.1}$$

This condition states that at no decision point h can a player i gain by deviating from π_i^* if after point h no other player j deviates from their optimal strategy π_j^*. Equilibria that satisfy this condition can be found using backward induction, however, von Neumann and Morgenstern [1] argue that backward induction makes the strong assumption of "persistent" rationality of the players.

3.4.1 The Chain Store Paradox and Irrational Players

Consider the table below which describes the payoffs of the chain store paradox expressed as a tuple (a, b), where a is the payoff of the store owner, and b is the payoff of the burglar. In this game, if the burglar breaks in and the store owner acquiesces, the payoff of the game is (-1,2), to the store owner and burglar, respectively. If the store owner chooses to fight, and the burglar still tries to break in,

		Store Owner	
		Acquiesce	Fight
Burglar	Break in	(-1,2)	(-5,-2)
	Do nothing	(0,0)	(-2,-1)

Table 3.1: The chain store paradox.

the payoff is (-5,-2). In this game the owner has a cost of maintaining a credible threat of choosing to fight, so if the burglar instead chooses to do nothing, the payoff of the game is (-2,-1). Finally, if the owner acquiesces and the burglar does nothing the payoff is (0,0).

If the Burglar breaks in, the optimal action for the store owner is to acquiesce as discussed above. However, if the store owner chooses to fight, then it would be optimal for the burglar to change strategy and do nothing, as this leads to a better outcome for the burglar. But if the owner knows the burglar is going to do nothing, it is better for the owner to stop maintaining the threat of choosing to fight and instead acquiesce. Now if the burglar knows that the owner is no longer willing to fight, they will choose to break in again, bringing us back to the start of this reasoning. This is the resultant paradox that occurs when trying to apply backward induction to the chain store paradox.

Selten [23] was the first to demonstrate that solutions determined by condition 3.1 can be difficult to accept as practical. Selten considers a version of the chain store paradox in which the owner of a chain store is threatened by repeated break-ins. When a break-in occurs, the store owner either acquiesces or fights. As we see in Table 3.1, acquiescing dominates fighting for the owner, however, breaking in only weakly dominates doing nothing for the burglar. Backward induction suggests that the store owner should always acquiesce in any given round when considered in isolation. However, we would expect the store owner to behave aggressively (choose to fight) at the beginning of the repeated game with the aim of making it undesirable for the burglars to break-in in future rounds.

If the burglar's expectation is that the store owner is rational, then as long as the store owner continues to choose to fight, the burglar will also be choosing a sub-optimal action by continuing to break in. If the burglar instead realises that the assumption that the store owner will behave rationally has been violated, and believes that the store owner will continue to be irrational, then the burglar will prefer to do nothing, which benefits the store owner in the long run of a repeated game.

The cause of the chain store paradox is the assumption of persistent rationality that underlies the self-enforcing condition 3.1. This assumption forces the players to believe that their opponent is still rational even after observing the opponent choosing irrational moves. Condition 3.1 assumes that all players behave rationally and expect all other players to behave rationally, however, once a player has demonstrated that they are irrational, this assumption is violated and is no longer appropriate.

In this research report we will explore what happens when players behave irrationally and what effect that has on the game outcome as well as the ability to estimate the skill of an irrational player. We will also discuss whether it is still appropriate for a player to assume their opponent is behaving rationally even after observing irrational play.

3.4.2 Trembling Hand Perfect Equilibria

Van Damme [21] notes that Selten [22] extended the arguments leading to condition 3.1 beyond games with perfect information. Consider a sub-game σ of game g. The expected payoff of the optimal

strategy π_i^* in sub-game σ depends only on what π_i^* prescribes in σ. We will denote this sub-strategy as $\pi_{i,\sigma}^*$. Selten then defines a sub-game perfect equilibrium as an equilibrium π_i^* of g that induces a Nash equilibrium $\pi_{i,\sigma}^*$ in each sub-game σ of g. Selten notes that since every equilibrium of a sub-game of a finite game can be extended to an equilibrium of the overall game, it follows that every finite extensive form game has at least one sub-game perfect equilibrium.

To eliminate non-self-enforcing equilibria, Selten [24] proposes that looking at complete rationality as a limiting case of incomplete rationality. The assumption here is that players can make mistakes with a small probability. These equilibria in these circumstances are called trembling hand perfect equilibria.

For an extensive form game g, assume that at each information set $h \in H_i$, player i will suffer from "momentary insanity" [24] and make a mistake with probability $\epsilon_h > 0$. It is assumed that $\epsilon_h > 0$ does not depend on the intended action at h. Selten defines a modified version of condition 3.1 that produces equilibrium strategies $\bar{\pi}_i$ under these "trembling hand" conditions.

In this research report we introduce the idea that the probabilistic outcomes of a player's actions are dependent on that player's skill. This allows us to embed a notion of skill into the game and to provide positive support to each possible sub-game. By doing this, we do not need to add the additional "trembling hand" conditions that Selten uses to provide support to all sub-games.

3.4.3 Sequential Equilibria

Kreps and Wilson [25] propose to eliminate irrational behaviour in a different way to Selten by explicitly specifying beliefs at each information set h so that posterior expected payoffs can always be computed. Players are made to conform with Bayesian decision theory by requiring that they are able to produce a probability distribution on the nodes in the information set h that represent the players' uncertainty. Players' beliefs need to be consistent with the strategies played and should respect the structure of the game. Kreps and Wilson achieve this by deriving the beliefs from a sequence of completely mixed strategies that converges to the desired strategy profile.

Kreps and Wilson define system of beliefs μ as a map that assigns each information set $h \in \cup_i H_i$ a probability distribution μ_h on the nodes in that set. The system of beliefs μ is said to be consistent with the strategy profile π_i^* if there exists a sequence π_i^k of completely mixed behaviour strategies with $\pi_i^k \to \pi_i^*$ as $k \to \infty$ such that:

$$\mu_h(s) = \lim_{k \to \infty} P_{\pi_i^k}(s|h) \text{ for all } h, x, \tag{3.2}$$

where $P_{\pi_i^k}(s|h)$ is the conditional probability that s is reached given that h is reached and π_i^k is played.

The strategy profile π_i^* is said to be sequentially rational given μ if:

$$V_{i,h}^{\mu}(\pi_i^*) \geq V_{i,h}^{\mu}(\pi_i^* \backslash \pi_i) \text{ for all } i, h, \pi_i \tag{3.3}$$

An assessment (π_i^*, μ) is said to be a sequential equilibrium if μ is consistent with π_i^* and if π_i^* is sequentially rational given μ. "Perfect equilibria require ex post optimality approaching the limit, while sequential equilibria only requires this at the limit" [25]. Thus, if π_i^* is perfect, then there exists some μ such that (π_i^*, μ) is a sequential equilibrium. However, if (π_i^*, μ) is a sequential equilibrium, it is not necessarily true that π_i^* is perfect. Van Damme [21] notes that the difference between perfect

and sequential equilibrium is marginal and that for almost all most games the two concepts yield the same outcomes.

In this research report we explore a solution to our proposed game that is similar, but not quite the same as Kreps and Wilson's concept of sequential equilibria. In our game, players will have beliefs about their opponent's skill, which will inform a mixed strategy to be followed by the player. In our game, a player's strategy is a function of their skill and their belief about their opponent's skill. However, the player's only reason about the observed play and possible strategy being followed by their opponent. The player's do not reason about the other player's reasoning. Also, for the sake of tractability in estimating skill, the strategy of the opponent is held constant during estimation.

This differs from the beliefs in the theory of Kreps and Wilson whose definition of player beliefs require players to reason about their opponent's reasoning. Kreps and Wilson do not provide any method or algorithm to find the sequential equilibria that they prove exists in their framework. One of our aims in this research report is to provide an algorithm for finding a solution to our proposed game.

In our proposed framework, the player observes play from their opponent and updates their beliefs using a Bayesian update on a prior distribution of the opponent's skill. As a player's belief distribution localises on the true skill of the opponent, we expect that their mixed strategy will converge to the optimal strategy that an oracle with perfect knowledge of the opponent's skill would have followed.

Chapter 4

Research Objectives and Methodology

This research report considers a subset of adversarial, two-player, zero-sum Markov games where skill is involved. In our framework we define transition dynamics of the Markov game as not only a function of the state and action choice of the rat player but also as a function of the skill of the rat player. We embed the concept of skill into the game by making the transition dynamics dependent on skill which requires players to reason about skills when selecting actions for themselves and when modelling their opponent. This gives us a mechanism by which players can estimate the skill of their opponent by observing transitions in the game. Players can determine the likelihood of observing an opponent's transition based on the assumed skill of the opponent.

By using adversarial, two-player, zero-sum Markov games where skill is involved, we are able to explore opponent modelling and skill estimation in these games. Similarly, the transition dynamics of the cat would also depend on the skill of the cat, but this skill is unknown to the rat.

In order for the rat to make use of tools like Expectiminimax [18] to derive an optimal policy and value function, the rat needs to estimate, or at least make an assumption about, the unknown skill of the cat, to determine the cat's transition dynamics. The optimal policy that the rat arrives at using Expectiminimax depends on the rat's assumption about the cat's skill. It also depends on whether the rat chooses to make a static assumption or chooses to estimate the cat's skill based on observed play.

In this research report we make the following hypotheses about the chosen subset of Markov games which rely on skill:

1. Incorrect assumptions cause harm: The expected reward obtained by a player making an incorrect assumption about their opponent's skill is always less than or equal to the expected reward when making the correct assumption about the opponent's skill.

2. Players can estimate skills from observed play: When players use Bayesian inference in estimating their opponent's skill their posterior distribution concentrates probability mass around the true value of the opponent's skill when sufficient play is observed.

3. Skill estimation is valuable: When a player uses Bayesian estimates of their opponent's skill to inform their own policy, the expected reward for that player converges to the expected reward of a player that is given perfect knowledge of their opponent's skill.

We now describe the experimental setup and methodology that is used to explore a subset of discrete, skills-based Markov games that will assist us in testing these hypotheses. In Section 4.1

we provide details of the game environment and how it is constructed, how the transition dynamics are defined as a function of skill, and what the win conditions for each player are. We also provide illustrations of the instances of our game environment that are used in our experiments. In Section 4.2 we describe a modified version of the Expectiminimax algorithm that can be used to compute the expected reward function for players in our game environment. Finally, in Section 4.3 we describe how Bayes Theorem is used in estimating skills based on observed play.

4.1 Markov Game Environment

As a starting point we wish to encode some notion of skill into a Markov game. This is desirable as the concept of player skills are present in many games. We desire a game environment in which players with higher skill have either more actions available to them than lower skilled players, or being able to perform certain actions more consistently than lower skilled players. In these Markov games with skill, players generally demonstrate more conservative play against higher skilled opponents as the chances of losing the game are increased, while being more aggressive against lower skilled opponents where winning is more certain. In order to embed this feature of skill in our Markov game we choose to encode skills into the stochastic dynamics of the game.

Before we define the game environment we first need to be more explicit about the notion of skill.

1. Assume we have a set A of all actions in the game.
2. There is a sequence of action sets $A_0 \subseteq A_1 \subseteq ... \subseteq A_k$.
3. We use the term "skill" as shorthand to refer to the index of the action sets above.
4. The action $a \in A_i$ can be performed successfully with probability P_i such that $P_0 \leq ... \leq P_i ... \leq P_k$.

Notice that for this definition of skill, higher skill values indicate players with more actions available in their action set. We also note that for any given action, a, players with higher skill are able to perform that action with a higher probability of success.

Higher skilled players are more likely to identify and exploit opportunities to gain an advantage over their opponents. This can result in a higher probability of winning or gaining an advantage in the game. Conversely, lower skilled players may be less likely to identify and exploit these opportunities, which can result in a lower probability of winning or gaining an advantage.

We explore these ideas in a custom domain which has two players (a cat and a rat) competing in a pursuit-evasion game on a directioned, weighted graph $G = (N, E, w)$ with nodes N, edges E, and edge weights $w \in [0, 1]$. Using the Markov game framework set out in Section 2.2 we introduce the notion that each player i has a skill $c_i \in [0, 1]$. In our framework we consider a discretisation of $[0, 1]$ when choosing initial values for c_i so that they can be mapped to the discrete, finite sequence of action sets $A_0 \subseteq A_1 \subseteq ... \subseteq A_k$, however the framework is capable handling continuous values of c_i.

Two nodes are selected from the graph as starting nodes for each player. The players then take alternating turns (starting with the rat) to move around the graph. On their turn, player i chooses an edge $e_i \in \{e \in E \mid e = (s_i(t), s'_i(t+1))\}$ on the graph along which it attempts to transition from state $s_i(t)$ to new state $s'_i(t+1)$. The action of transitioning from state $s_i(t)$ to state $s'_i(t+1)$ along edge e_i is successful with probability $P_i^e(s, s')$. If the action is unsuccessful, with probability $1 - P_i^e(s, s')$,

player i transitions to state $s_i(t+1)$ (i.e. the player remains in the current state and has effectively lost a turn).

For our game environment we also define a set of goal states $S_i^{\text{goal}} \subseteq S$ for player i. If player i finds themselves in one of the goal states, $s_i \in S_i^{\text{goal}}$ then player i is declared the winner of the game.

4.1.1 Transition Dynamics

We have defined a transition function such that if the player has skill greater than or equal to the edge weight, then that player may transition across that edge with the least uncertainty about the action succeeding. If the player has skill less than that of the edge weight, the transition probability is reduced. Players cannot use edges which do not exist in the game and, as such, the transition probability between states with no connecting edge is be zero. Players are allowed to remain on their current node with probability 1 which allows players to effectively pass their turn if no other move is advantageous to them.

We choose a transition function that decreases as the difference between the chosen edge's weight and the player's skill increase. We refer to this difference as the skill differential $w_e - c_i$ where w_e is the weight of edge $e \in E$. This transition function has the desirable effect that the more positive the skill differential (i.e. the edge weight is higher than the player skill), the less likely that player succeeds in taking the chosen edge. Note that we have chosen not to penalise or reward negative skill differentials (i.e. where the player skill is greater than or equal to the edge weight required). The choice of linearity is arbitrary as the game design can use any continuous transition function. However, it is recommended that the function be monotonically decreasing as a function of the skill differential in order to preserve the intuition that higher skilled players succeed at actions more often than lower skilled players.

We introduce a parameter into our game that increases the probability of action failure in the transition dynamics. This failure factor, $\kappa \in [0, 1]$, introduces additional uncertainty for players of all skills by bounding the transition probabilities above to a value of $1 - \kappa$.

This increased probability of actions failing ensures that all players experience a minimum level of uncertainty that their chosen action results in the desired outcome. Without the additional chance of failure, players with sufficient skill would be able to execute all available actions with a certain outcome, making the dynamics deterministic. This provides additional flexibility to our framework to cater for games where there is always an element of uncertainty in the outcome of a chosen action regardless of skill.

In our framework, players are precluded from choosing actions which require them to take edges that do not exist. All players are thus assumed to take only legal actions.

With the above considerations we can now specify our transition probability function as:

$$P_i^e(s, s', c_i, w_e, \kappa) = \begin{cases} 1 & \text{if } s = s', \\ 1 - \max(w_e - c_i, \kappa) & \text{if } s \neq s' \text{ and } e \in E, \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

From Equation 4.1 it is clear that transition uncertainty is driven by the value of $\max(w_e - c_i, \kappa)$. One action available to players is to remain in their current state with certainty. Alternatively, players may choose to transition to a different state along an available edge with some transition probability

that is less than 1. In Figure 4.1 we plot the transition probability function given by Equation 4.1, as a function of the skill differential $w_e - c_i$, for different values of κ.

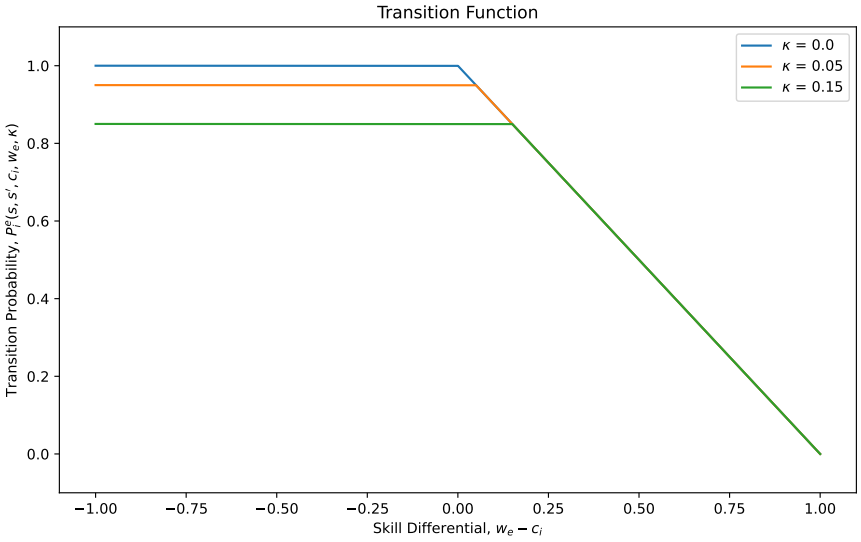


Figure 4.1: Transition probability function for different values of the failure factor κ.

In Figure 4.1 we observe that the transition probability for $\kappa = 0$ is either flat or decreasing for increasing skill differentials, with a point of non-differentiability where $w_e = c_i$. For increasing κ, the transition probability function shifts downwards and the inflection point moves to the right. This increase in κ provides a marginal benefit to some lower skilled players since a small positive skill differential still results in the same transition probability as a player with negative skill differential, so long as for the lower skilled player $w_e - c_i < \kappa$.

4.1.2 Win Conditions and Scoring

Here we define a set of mutually exclusive win conditions for our game as follows:

- If $s_{\text{rat}}(t) \in S_{\text{rat}}^{\text{goal}}$ at any time t (i.e. the rat reaches one of the goal states in S_{rat}) then the rat is declared the winner, the game ends, and the rat receives a reward of 1 and the cat receives a reward of -1.
- If $s_{\text{cat}}(t) = s_{\text{rat}}(t)$ at any time t (i.e. the cat catches the rat) then the cat is declared the winner, the game ends, and the cat receives a reward of 1 and the rat receives a reward of -1.
- If $t = T$ and neither of the above conditions have been satisfied the game ends in a tie and the rat is given a reward of $\beta \in [-1, 1]$ and the cat receives a reward of $-\beta$.

In the analysis that follows, we consider the game score to be the reward from the rat's perspective. If the rat wins the game score is 1, if the cat wins the game score is -1, and if the game ends in a tie, the score is β. Thus we can think of the rat as the maximising player and the cat as the minimising player in an Expectiminimax framework. We also consider the case where the third win condition is modified to declare a draw when $t = T$ and neither of the first two win conditions have been met. In this setup, the score for each player is zero if the game is a draw.

The graph G in this game environment can be generated to represent an arbitrary game where the number of nodes and edge/weight configurations can be specified. The game environment can therefore produce a family of Markov games that have infinite variety. The game environment also explicitly encodes the notion of skill as described in Section 4.1.1.

4.1.3 Game Instances

In Figure 4.2 we have one instance of such a game where node 0 is the goal for the rat ($S_{\text{rat}} = N_0$) that fulfills the first win condition.

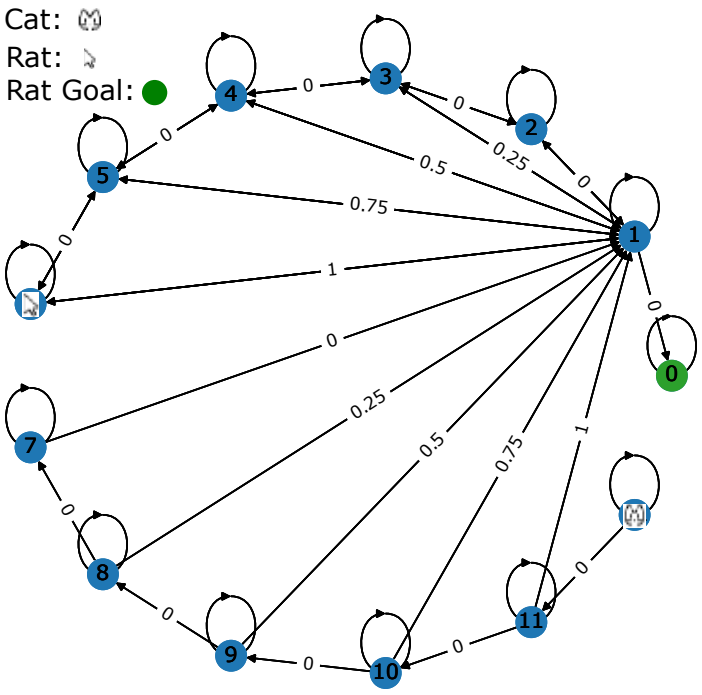


Figure 4.2: Contrived cat and rat game instance.

We have contrived a game in Figure 4.2 where the cat is represented by the image of a cat's face, the rat is represented by the image of a mouse pointer, and the goal for the rat is represented by the green node, in this instance node 0. We also have nodes 1 to 12 in blue that represent the states in our game. At the start of this particular game the rat occupies node 6 and the cat occupies node 12. The edges in the graph are directed and each edge has the weight value displayed in the middle of the edge. Some edges are bidirectional, for example between nodes 1 and 2, meaning players can travel from node 1 to 2 and from node 2 to 1. Some edges are only one-directional, for example between nodes 1 and 10, meaning players can only travel in one direction, from node 10 to 1.

In this contrived game, the higher each player's skill, the shorter the path each player can take with maximal probability to reach the rat's goal. This is because players need a skill greater than or equal to the weight of the edge to take that edge without a skill differential penalty. For each player, the optimal policy needs to balance the path length to the rat goal with the probability of being able to successfully take that path in order to achieve a win condition.

Due to the nature of skills in this game, and their effect on transition dynamics, we may observe some interesting play for lower skilled players based on the assumptions players make about their opponent's skill. A lower skilled player may choose to risk taking an edge above its skill, resulting in a

lower probability of winning, over taking edges that are equal to or below its skill, resulting in a high probability of losing the game.

With this game environment we have the ability to randomly generate directed graphs with an arbitrary number of nodes and edges. While the possibilities for different games are endless, we have imposed some structure on our randomly generated graphs to encourage play from the players that allows us to better test our hypotheses. This structure consists of a ring of nodes with each note connected to the next and preceding node. We then randomly generate a chosen number of shortcut edges which connect non-successive nodes to each other. Finally, we generate a goal node with a single edge connecting it to one of the nodes in the ring. The edges that return the player to the same node have weights set to zero. All other edge weights are generated randomly.

In Figure 4.3 we present two such possible randomly generated games, one with a small number of nodes and one with a large number of nodes.

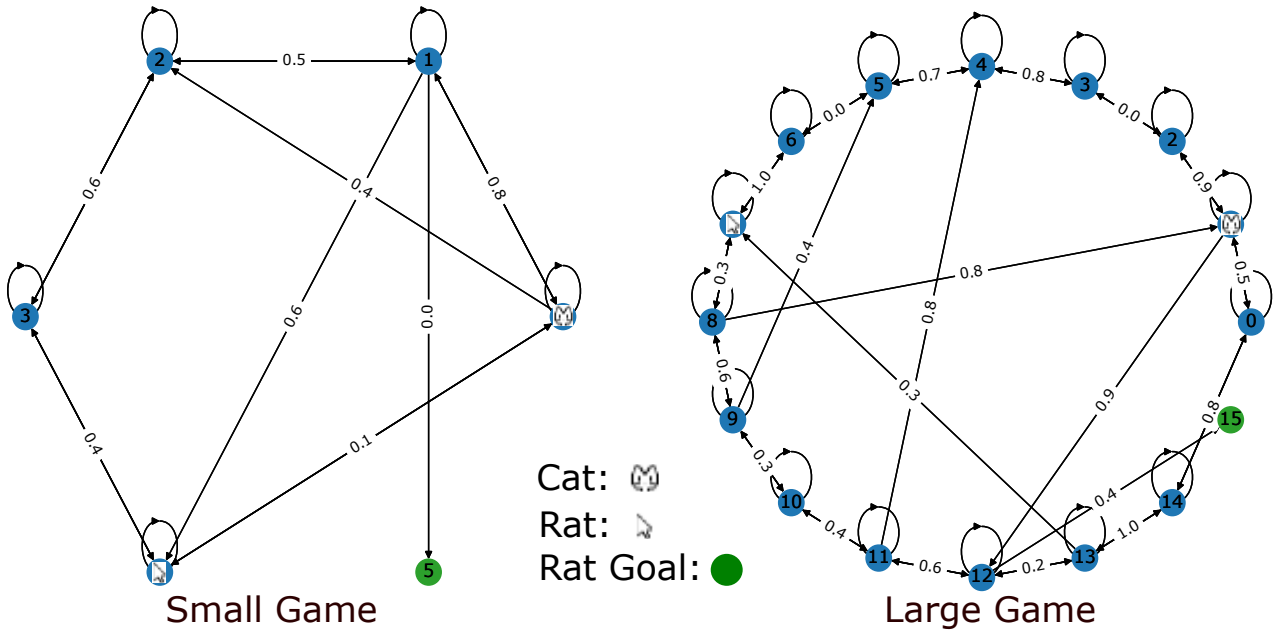


Figure 4.3: Random instances of small and large sized games.

With this game environment we can now formulate our game evaluation and policy iteration algorithms. These allow us to assess the performance of players with various skills and assumptions about their opponent's skills.

4.2 Game Evaluation

Let us assume that for a known game graph, the rat knows their own skill but not that of their opponent, the cat. Now consider a rat player with some a priori assumption about their opponent's skill $\hat{c}_{\text{cat}}$. The rat is interested in playing to maximize their discounted expected reward. Each player is assumed to use the same discount factor $\gamma \in (0, 1]$ at each time step. When $\gamma < 1$, players prefer reaching their respective win conditions sooner rather than later and tolerate more uncertainty in achieving the desired reward. When $\gamma = 1$, players prefer a more certain reward over a less certain reward even if it takes more moves to reach the more certain reward.

We construct an expected value matrix from the rat's perspective $V_{\text{rat}}(s, t, \hat{c}_{\text{cat}}, \kappa, \gamma, \beta)$ in each

state s and at each time t using the rat's a priori assumption about the cat's skill $\hat{c}_{\text{cat}}$. Starting at the terminal game time $t = T$, we initialise the expected value matrix $V_{\text{rat}}(s, T, \hat{c}_{\text{cat}}, \kappa, \gamma, \beta)$ with the terminal scores of the game described in Section 4.1 as follows:

$$V_{\text{rat}}(s, T, \hat{c}_{\text{cat}}, \kappa, \gamma, \beta) := \begin{cases} 1 & \text{if } s_{\text{rat}} = N_{\text{goal}}, \\ -1 & \text{if } s_{\text{cat}} = s_{\text{rat}}, \\ \beta & \text{otherwise.} \end{cases} \tag{4.2}$$

In Equation 4.2 we see that the reward at time T is 1 if the rat wins by reaching the goal node, the reward is -1 if the cat catches the rat, and the reward is β for every other state of the game. We can think of β as the reward to the rat if the game is a tie, with the cat receiving a reward of $-\beta$.

In the chapters that follow we consider two values for β. In one set of experiments, we choose $\beta = 1$, which rewards the rat for evading the cat for T turns. In another set of experiments we choose $\beta = 0$, which rewards neither player for allowing the game to reach T turns.

The rat may also receive rewards at any other time t during the game. If the rat reaches the goal node at any time t the rat receives a reward of 1. If the cat catches the rat at any time t the rat receives a reward of -1. We express this formally as:

$$R^{\pi^*_{\text{rat}}}_{\text{rat}}(s, s') := \begin{cases} 1 & \text{if } s'_{\text{rat}} = N_{\text{goal}}, \\ -1 & \text{if } s'_{\text{cat}} = s'_{\text{rat}}. \end{cases} \tag{4.3}$$

Using the rewards specified in Equation 4.2 and iterating backwards in time for each player it is possible to update the rat's expected reward matrix $V_{\text{rat}}(s, t, \hat{c}_{\text{cat}}, \kappa, \gamma, \beta)$ for previous time steps. However, we have introduced modified transition probabilities $P^e_{\text{rat}}(s, s', c_{\text{rat}}, w_e, \kappa)$ which players need to account for when computing the expected reward. The rat also need to use their assumption, $\hat{c}_{\text{cat}}$, about the cat's skill to determine the transition dynamics of the cat, $P^e_{\text{cat}}(s, s', \hat{c}_{\text{cat}}, w_e, \kappa)$.

We can modify the Expectiminimax algorithm from Section 2.3 to compute the matrix of expected rewards $V_{\text{rat}}(s, t, c_{\text{rat}}, \hat{c}_{\text{cat}}, \kappa, \gamma, \beta)$ for the rat and its skill assumption dependent optimal policy $\pi^*_{\text{rat}}(s, t, c_{\text{rat}}, \hat{c}_{\text{cat}}, \kappa, \gamma, \beta)$.

We can compute the expected reward for the rat recursively for times $t = \{T, T-1, \ldots, 2, 1\}$ as follows:

$$\begin{aligned} V_{\text{rat}}(s, t-1, c_{\text{rat}}, \hat{c}_{\text{cat}}, \kappa, \gamma, \beta) \;=\; & \sum_{s'} P^{\hat{\pi}^*_{\text{cat}}}_{\text{cat}}(s, s', \hat{c}_{\text{cat}}, w_{\hat{\pi}^*_{\text{cat}}}, \kappa) \times \\ & \min[R^{\hat{\pi}^*_{\text{cat}}}_{\text{cat}}(s, s'), \gamma V_{\text{rat}}(s', t, c_{\text{rat}}, \hat{c}_{\text{cat}}, \kappa, \gamma, \beta)], \end{aligned} \tag{4.4}$$

$$\hat{\pi}^*_{\text{cat}}(s, t-1, c_{\text{rat}}, \hat{c}_{\text{cat}}, \kappa, \gamma, \beta) = \underset{a}{\text{argmin}}\{P^a_{\text{cat}}(s, s', \hat{c}_{\text{cat}}, w_a, \kappa) \times V_{\text{rat}}(s, t-1, c_{\text{rat}}, \hat{c}_{\text{cat}}, \kappa, \gamma, \beta)\}, \tag{4.5}$$

$$\begin{aligned} V_{\text{rat}}(s, t-2, c_{\text{rat}}, \hat{c}_{\text{cat}}, \kappa, \gamma, \beta) \;=\; & \sum_{s'} P^{\pi^*_{\text{rat}}}_{\text{rat}}(s, s', c_{\text{rat}}, w_{\pi^*_{\text{rat}}}, \kappa) \times \\ & \max[R^{\pi^*_{\text{rat}}}_{\text{rat}}(s, s'), \gamma V_{\text{rat}}(s', t-1, c_{\text{rat}}, \hat{c}_{\text{cat}}, \kappa, \gamma, \beta)], \end{aligned} \tag{4.6}$$

$$\pi^*_{\text{rat}}(s, t-2, c_{\text{rat}}, \hat{c}_{\text{cat}}, \kappa, \gamma, \beta) = \underset{a}{\text{argmax}}\{P^a_{\text{rat}}(s, s', c_{\text{rat}}, w_a, \kappa) \times V_{\text{rat}}(s, t-2, c_{\text{rat}}, \hat{c}_{\text{cat}}, \kappa, \gamma, \beta)\}. \quad (4.7)$$

Notice that the equations above are from the rat's perspective, hence we are updating the expected reward for the rat $V_{\text{rat}}(s, t, c_{\text{rat}}, \hat{c}_{\text{cat}}, \kappa, \gamma, \beta)$ under the assumption $\hat{c}_{\text{cat}}$ of the cat's skill and the rat's corresponding optimal policy $\pi^*_{\text{rat}}(s, t-2, c_{\text{rat}}, \hat{c}_{\text{cat}}, \kappa, \gamma, \beta)$.

We have a similar algorithm for computing the expected reward matrix $V_{\text{cat}}(s, t-1, c_{\text{cat}}, \hat{c}_{\text{rat}}, \kappa, \gamma, \beta)$ and optimal policy $\pi^*_{\text{cat}}(s, t-1, c_{\text{rat}}, \hat{c}_{\text{rat}}, \kappa, \gamma, \beta)$ from the perspective of the cat.

Let $V^*_{\text{rat}}(s, t, c_{\text{rat}}, c_{\text{cat}}, \hat{c}_{\text{rat}}, \hat{c}_{\text{cat}}, \kappa, \gamma, \beta)$, which we will abbreviate to $V^*_{\text{rat}}(\cdot)$ going forward, be the expected reward function of the rat when both players execute their optimal policies under their assumptions about their opponent's skill. We have the optimal cat policy $\pi^*_{\text{cat}}(s, t-1, c_{\text{cat}}, \hat{c}_{\text{rat}}, \kappa, \gamma, \beta)$ for odd t and the optimal rat policy $\pi^*_{\text{rat}}(s, t-2, c_{\text{rat}}, \hat{c}_{\text{cat}}, \kappa, \gamma, \beta)$ for even t. We can use the Expectiminimax algorithm from Section 2.3 and the true transition dynamics of the players to compute the expected reward matrix $V^*_{\text{rat}}(\cdot)$ for all states, times, player skills, and opponent skill assumptions. Where there are multiple optimal actions for a given state and time the player selects an optimal action at random. We initialise $V^*_{\text{rat}}(\cdot)$ at time T with the values from Equation 4.2.

The algorithm described above can now determine the optimal policy for each player based on their skill and their a priori assumption about their opponent's skill. With the optimal policy for each player, and the expected reward matrix, we can compare the resulting optimal policies $\pi^*_{\text{rat}}(s, t, c_{\text{rat}}, \hat{c}_{\text{cat}}, \kappa, \gamma, \beta)$ for different choices of $\hat{c}_{\text{cat}}$ and measure the change in $V^*_{\text{rat}}(\cdot)$. We can also determine the cost of optimism vs pessimism in the choice of skill assumption by measuring the changes in the resulting discounted expected reward.

4.3 Skill Estimation

In this section we describe how the rat can estimate the cat's skill based on observed play and vice versa. It is important to note that players observe the resulting transition of their opponent in the game and not the action choice the opponent made. This is important because if, for example, the rat know the cat's chosen action, then it becomes much easier for the rat to estimate the cat's skill, $\hat{\pi}^*_{\text{cat}}(s, t, c_{\text{rat}}, \hat{c}_{\text{cat}}, \kappa, \gamma, \beta)$, from the transition probability in Equation 4.1, where $i = \text{cat}$.

Due to not knowing the opponent's actions, we define the finite set of observable moves $M = \{m = (s, s') | s, s' \in S\}$. These moves are observed by players, or interested third parties, in the absence of any information about what action was taken by the transitioning player.

In Section 4.2 the rat made an a priori assumption about the skill of the cat. We may relax the assumption that $\hat{c}_{\text{cat}}$ is constant throughout the game and instead have an adapted estimate of the cat's skill $\hat{c}_{\text{cat}}(t)$ based on observed play. Since the optimal policy $\pi^*_{\text{rat}}(s, t, c_{\text{rat}}, \hat{c}_{\text{cat}}, \kappa, \gamma, \beta)$ is a function of the assumption $\hat{c}_{\text{cat}}$, it is desirable to estimate $\hat{c}_{\text{cat}}$ and thus compute an adapted optimal policy for the rat based on observed play of the cat.

However, the cat's policy $\pi^*_{\text{cat}}(s, t, c_{\text{cat}}, \hat{c}_{\text{rat}}, \kappa, \gamma, \beta)$ is a function of their skill, c_{cat}, which is known to them, and their a priori assumption about the rat's skill $\hat{c}_{\text{rat}}$. Therefore, each move the cat makes may provide information about both their true skill c_{cat} and their a priori assumption $\hat{c}_{\text{rat}}$ about the rat's skill.

We can modify the game evaluation algorithm to accommodate the new adapted skill estimate $\hat{c}_{\text{cat}}(t)$. Assume we have observed the following move history $\mathcal{M}_\tau = \{m_0, m_1, \ldots, m_\tau\}$ made by the

cat from $t = 0$ until $t = \tau$. The rat can estimate the distribution of the cat's skill and their a priori assumption about the rat's skill given $\mathcal{M}_\tau$ using Bayesian Inference in the following way:

$$P(\hat{c}_{\text{cat}}(\tau), \hat{c}_{\text{rat}} | \mathcal{M}_\tau) = \frac{P(\mathcal{M}_\tau | \hat{c}_{\text{cat}}(\tau), \hat{c}_{\text{rat}}) P(\hat{c}_{\text{cat}}(\tau), \hat{c}_{\text{rat}})}{P(\mathcal{M}_\tau)}. \tag{4.8}$$

Where,

- $P(\hat{c}_{\text{cat}}(\tau), \hat{c}_{\text{rat}} | \mathcal{M}_\tau)$ is the posterior joint distribution of the cat's skill and a priori assumption;
- $P(\mathcal{M}_\tau | \hat{c}_{\text{cat}}(\tau), \hat{c}_{\text{rat}})$ is the likelihood which we can construct from the transition function given in Equation 4.1;
- $P(\hat{c}_{\text{cat}}(\tau), \hat{c}_{\text{rat}})$ is the prior joint distribution of the cat's skill and a priori assumption; and,
- $P(\mathcal{M}_\tau)$ is the evidence.

In particular, the likelihood can be expressed as a function of the transition probabilities from Equation 4.1 as follows:

$$P(m | \hat{c}_{\text{cat}}(t), \hat{c}_{\text{rat}}, \hat{\pi}^*_{\text{cat}}) = \begin{cases} P^m_{\text{cat}}(s, s', \hat{c}_{\text{cat}}, w_m, \kappa)/n(\hat{\pi}^*_{\text{cat}}) & \text{if } m \in \hat{\pi}^*_{\text{cat}}, \\ \sum\limits_{e \in A^s_{\text{cat}}} [1 - P^e_{\text{cat}}(s, s', \hat{c}_{\text{cat}}, w_e, \kappa)/n(\hat{\pi}^*_{\text{cat}})] & \text{if } m \notin \hat{\pi}^*_{\text{cat}}, \end{cases} \tag{4.9}$$

where $n(\hat{\pi}^*_{\text{cat}})$ is the number of optimal actions that the rat believes is available to the cat in state s at time t. Intuitively, we can interpret Equation 4.9 in the following way:

- When the cat transitions to another state, the likelihood of that move is equal to the transition probability under the rat's assumption about the cat's skill, divided by the number of optimal actions that the rat believes are available to the cat; and,
- when the cat remains in the same state, the rat can deduce that the cat's chosen optimal action failed and therefore sum the failed transition probabilities across all optimal action choices.

We can thus compute the posterior distribution given observed moves from the cat $\mathcal{M}_t$. This posterior distribution can be updated either move-by-move or in batches. We have made the choice that one batch constitutes the moves made by the cat in one completed game. This allows us to simulate a full game to completion and use the recorded move histories of each player in our Bayesian estimator.

For a batch update of observed moves $\mathcal{M}_\tau = \{m_0, m_1, \ldots, m_\tau\}$, we have the likelihood function:

$$P(\mathcal{M}_\tau | \hat{c}_{\text{cat}}(\tau), \hat{c}_{\text{rat}}, \hat{\pi}^*_{\text{cat}}) = \prod_{i=0}^{\tau} P(m_i | \hat{c}_{\text{cat}}(i), \hat{c}_{\text{rat}}, \hat{\pi}^*_{\text{cat}}) \tag{4.10}$$

It should be noted that there are analogous equations for the cat estimating the skill and a priori assumption of the rat too.

We have chosen to begin the estimation using a uniform prior. While other choices of prior distribution may be considered, we chose a uniform prior since it does not bias the player in any way before it has observed moves from the opponent. Experiments involving different choices of prior distributions is beyond the scope of this research report.

Once we can estimate the opponent's skill and a priori assumption, we can analyse the resulting optimal policies and compare them with those generated by constant, unchanging assumptions about the opponent's skill. We can also analyse the resulting expected reward matrices under skill estimation and constant, unchanging assumptions about the opponent's skill to determine the cost of estimation.

We conduct experiments to observe the effects of the failure factor κ on the estimation of skills. We expect that the increase in transition failures leads to more explanations for the observed play, resulting in the skill estimates converging more slowly to the true skill of the opponent.

We also consider what happens to the skill estimation when opponents behave randomly when being observed. We note that playing randomly is considered non-strategic and would result in worse expected reward for players employing this tactic. Random players are assumed to select actions at random from the set of available actions with equal probability. To incorporate these random players into our estimation model we do require one restriction on their actions: random players may not chose an action that has non-zero probability of resulting in an immediate loss if the action is successful.

Backward induction and reinforcement learning are both approaches used in decision-making processes, but they differ in their fundamental principles and applications. While backward induction is well-suited for scenarios with known rules and optimal decision-making, reinforcement learning excels in environments with uncertainty and incomplete information, adapting to changing conditions through continuous learning. In the game environment we have described above, we embed skills into a game that has known rules, states, and transition dynamics. This makes our game environment better suited to employing backward induction rather than reinforcement learning as a tool for estimating skill.

In Chapter 5 we conduct experiments on the expected reward function to explore its properties and observe the effects of changes to parameters like κ, γ, and β on the outcomes of the game and the style of play observed from the players.

In Chapter 6 we conduct experiments to determine how accurately players can estimate the skill of their opponents. We also consider the performance gain to players conducting such estimation on their opponents and determine whether it is valuable for players to do this.

Chapter 5

The Expected Reward Function

In this chapter we conduct a series of experiments to explore the features of the expected reward function. We do this using our Markov game environment defined in Section 4.1 for which we can compute the rat's expected reward function, $V^*_{\text{rat}}(s, T, c_{\text{rat}}, c_{\text{cat}}, \hat{c}_{\text{rat}}, \hat{c}_{\text{cat}}, \kappa, \gamma, \beta)$, as described in Section 4.2. In the experiments that follow, we only consider the rat's expected reward function. Since the game is zero-sum, the cat's expected reward function would be the reflection of the rat's reward function about zero.

With these experiments we wish to test our hypothesis that the expected reward obtained by a player making an incorrect assumption about their opponent's skill is always less than or equal to the expected reward when making the correct assumption about the opponent's skill. We also wish to explore the effects of the parameters κ, γ, and β on the expected reward function.

In Section 5.1.1 we perform our first experiment which serves as a sanity check that our Expectiminimax algorithm for the game is functioning as intended. By simulating a number of rounds of the contrived game described in Figure 4.2, we compute the average score of the game and compare it with the expected reward computed by the closed-form Expectiminimax algorithm. We also compare the run-time of the closed-form solution for the expected reward function to the run-times of the simulated empirical reward function for increasing numbers of simulated games.

In Sections 5.1.2 and 5.1.3 we perform a second set of experiments in which we take 2-dimensional slices of the expected reward function along two sets of dimensions in order to be able to visualise and interpret how each of the skill and assumption parameters affect the expected reward. The first slice of the expected reward function we consider is along the cat skill and rat skill dimensions. The second slice of the expected reward function we consider is along the cat assumption and rat assumption dimensions.

From the graphs sliced along the skill dimensions we are able to observe that the expected reward is monotonically increasing for an increase in the rat's skill (maximising player). Likewise, the expected reward is monotonically decreasing for an increase in the cat's skill (minimising player).

From the graphs sliced along the assumption dimensions we are able to observe that the expected reward is concave along the rat assumption dimension, suggesting the rat (maximising player) attains the maximal reward (all else equal) when its assumption about the cat's skill is correct, and performs worse when it is incorrect. Similarly, the expected reward function is convex along the cat assumption dimension, suggesting that the cat (minimising player) attains the minimal reward (all else equal) when its assumption about the rat's skill is correct, and performs worse when it is incorrect.

The third experiment we conduct in Section 5.2 demonstrates the effects of the failure factor κ on

the expected reward function. We observe that the additional uncertainty in the transition dynamics introduced by choosing $\kappa > 0$ has the effect of reducing the overall magnitude of the players' expected rewards, as well as encouraging more conservative play from players in certain instances.

In Section 5.3 we conduct a fourth experiment which demonstrates the effects of players discounting the reward by choosing $\gamma < 1$. We observe that discounting not only has the effect of reducing the expected reward at the start of the game, but in some instances can induce some counter-intuitive behaviour in players.

Finally, in Section 5.4 we conduct our fifth experiment in which we set the reward for ties, β, to zero. This allows us to illustrate how the players (particularly the rat) behave in games where there is no reward for the rat if it manages to evade the cat for T turns. In all the experiments that follow we choose $T = 10$ which means players have a maximum of 10 turns each.

5.1 Computing Expected Rewards

5.1.1 Convergence of Empirical Results to Closed-Form Results

First we provide empirical evidence that the closed-form expected discounted reward function defined in Section 4.2 is correct. We do this by using the optimal policies π^*_{rat} and π^*_{cat} for each player to play a different number of rounds of the game. The average reward is computed across all games played. The empirical average reward should converge to the closed-form expected reward of the game as computed by the Expectiminimax algorithm.

We choose arbitrary values for $c_{\text{rat}}, c_{\text{cat}}, \hat{c}_{\text{rat}}$, and $\hat{c}_{\text{cat}}$ for the purposes of this convergence test. We also choose a failure factor of $\kappa := 0$, a discount factor of $\gamma := 1$, and the reward in the event of a tie to be $\beta := 1$. The choices for κ and γ enable us to compute the expected reward without discounting or additional uncertainty in the transition dynamics. The choice for β is arbitrary, however we assess the impact of a different choice of β in Section 5.4.

In Table 5.1 we present the closed-form expected reward with the empirical expected reward calculated over a different number of simulated games. These expected rewards are computed for the contrived game illustrated in Figure 4.2. The expected rewards and run-times for the empirical calculation were averaged across 10 independent runs. The error is calculated as percentage error relative to the closed-form expected reward.

Calculation Type	# Simulated Games	Expected Reward	Run-Time	Error
Closed-Form	N/A	0.14883	26.57ms	N/A
Simulated	10	0.04000	0.60ms	126.88%
Simulated	100	0.07400	5.50ms	50.28%
Simulated	1,000	0.11960	52.21ms	19.64%
Simulated	10,000	0.14430	534.57ms	3.04%
Simulated	100,000	0.14722	5,294.84ms	1.08%
Simulated	1,000,000	0.14928	50,752.30ms	-0.30%

Table 5.1: Computation of expected rewards for the contrived game.

From Table 5.1 we observe that to compute an empirical expected reward that is within approximately 1% of the closed-form expected reward, we are required to simulate on the order of 100,000

games with a run-time two orders of magnitude slower than the closed-form calculation. We also note that computing the expected reward empirically only provides the expected reward for the game for one initial state of the game. In contrast, the closed-form computation provides us with the expected reward for all possible starting states of the game. This makes the use of the closed-form calculation of the expected reward advantageous over the empirical calculation.

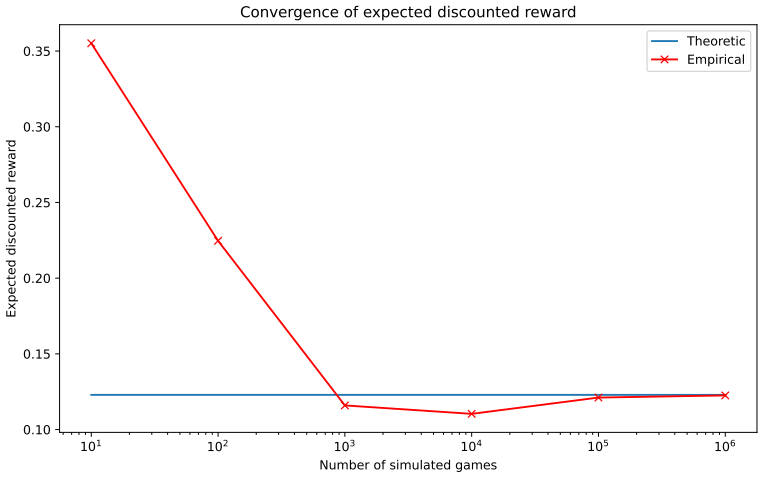


Figure 5.1: Discounted expected reward convergence for the contrived game.

In Figure 5.1 we illustrate the convergence of the empirical expected rewards to the theoretic expected reward. It is important to note that the x-axis in Figure 5.1 is on a log scale. We observe that the empirical expected rewards converge quickly at first, and then more slowly, with the inflection point in the graph occurring at 10,000 simulated games. The expected reward results that follow in this chapter are all calculated using the closed-form calculation.

5.1.2 Expected Reward Versus Skill

Now that we are comfortable that our modified Expectiminimax algorithm is supported by the empirical evidence, we can use it to compute the expected discounted reward for varying player skills and assumptions. For each game instance with a specified initial state, we can construct a 4-dimensional matrix of expected discounted rewards for varying $c_{\text{rat}}, c_{\text{cat}}, \hat{c}_{\text{rat}}$, and $\hat{c}_{\text{cat}}$ parameters. Unfortunately, we can only produce 3-dimensional plots of the expected discount reward function, meaning we need to hold two of the four parameters constant for any given plot. In this section we choose to plot the function for varying player skills while holding player assumptions constant.

In Figure 5.2 the graph on the left plots three expected reward function surfaces for the contrived game. For all three surfaces, the cat's assumption about the rat's skill is set to 0.5. Each surface in the plot represents a different rat assumption about the cat's skill of 0.3, 0.5, and 0.7 respectively for the blue, green, and red surfaces. We can observe that the expected reward surfaces have a positive slope along the rat skill axis, while the slope is negative along the cat skill axis. This makes intuitive sense since the rat is the maximising player and the cat is the minimising player. Each player is more likely to achieve their objective when their skill is high and their opponent's skill is low.

The graph on the right illustrates a very similar structure in the expected reward function. This time the rat's assumption about the cat's skill is held constant for all three surfaces, and each surface

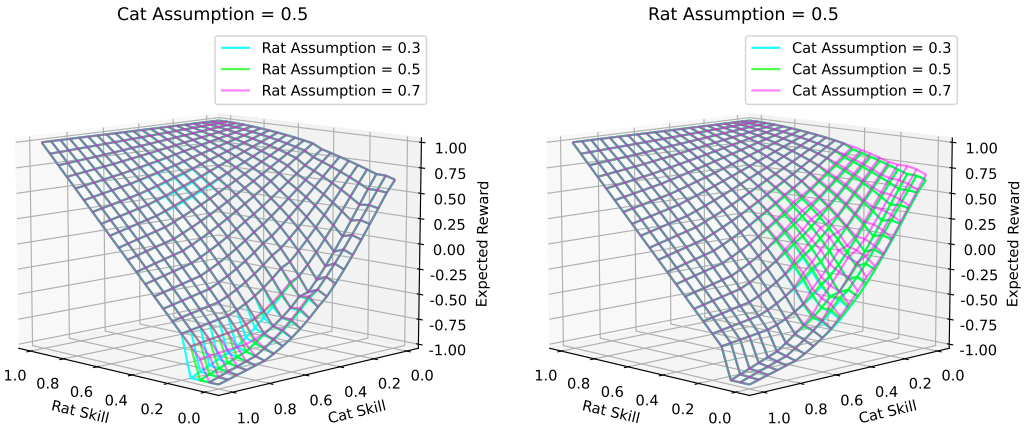


Figure 5.2: Expected reward vs player skill for the contrived game.

varies the cat's assumption about the rat's skill. Again we observe that the slope of the expected reward function is positive along the rat skill axis and negative along the cat skill axis.

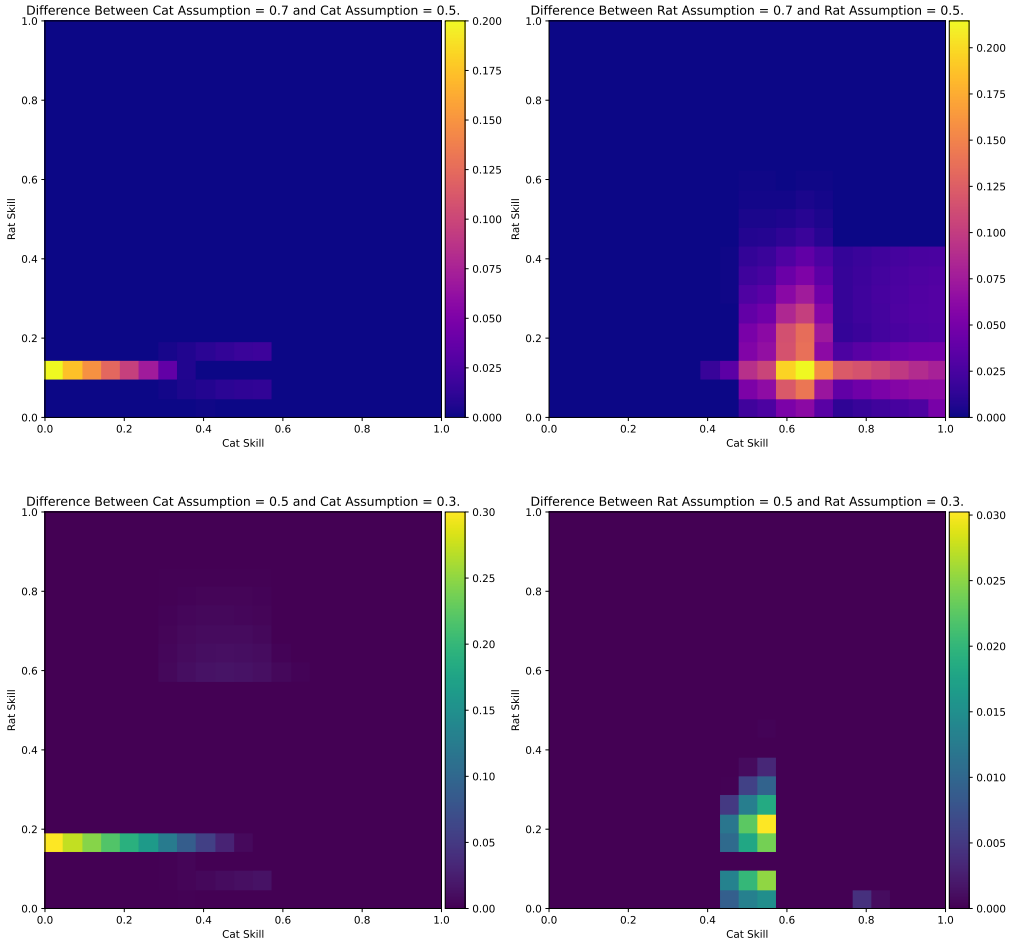


Figure 5.3: Heat-map of the differences between the surfaces of Figure 5.2.

From Figure 5.2 we observe that the expected reward function has a maximum of 1 where the rat's skill is the maximum of $c_{\text{rat}} = 1$. This makes sense given that the contrived game in Figure 4.2 is designed such that a rat with a skill of 1 always reaches the goal at node $N = 0$ before the cat,

regardless of the cat's skill. We also observe the expected reward function has a minimum of -1 where the cat's skill is the maximum of $c_{\text{cat}} = 1$ for some low values of the rat's skill.

It is interesting to note that there is a noticeable jump in the expected reward function where $c_{\text{rat}} < 0.2$ and $c_{\text{cat}} > 0.7$ on the green and cyan surfaces for the graph on the left. This feature is also present in all surfaces on the graph on the right. This is caused by a shift in the rat's policy from trying to reach the goal when $c_{\text{rat}} \geq 0.2$ as opposed to trying to evade the cat until the end of the game when $c_{\text{rat}} < 0.2$. The differences between the surfaces in Figure 5.2 are plotted in a heat-map in Figure 5.3 for ease of visualisation.

This shift in policy occurs in these specific cases because the rat holds the assumption that the cat's skill is lower than it truly is, i.e. the rat is being optimistic about the cat's skill. In these cases, when the rat's skill is sufficiently low, the expected reward as computed by the rat is maximised by evading the cat for as long as possible. Unfortunately for the rat, if the cat has a skill great enough, it catches the rat with certainty. This is why the expected reward function has a value of -1 where skills $c_{\text{cat}} = 1$, $c_{\text{rat}} < 0.2$, and the rat's assumption $\hat{c}_{\text{cat}} \leq 0.5$.

If we set the rat's assumption about the cat's skill to match the reality in this case, i.e. $\hat{c}_{\text{cat}} = 1$, then we observe in Figure 5.4 that the jump no longer appears in that region of the expected reward function. This is because the rat knows that it cannot evade the cat until the end of the game under its new assumption about the cat's skill. Thus the rat attempts to reach the goal as quickly as possible, regardless of its own skill.

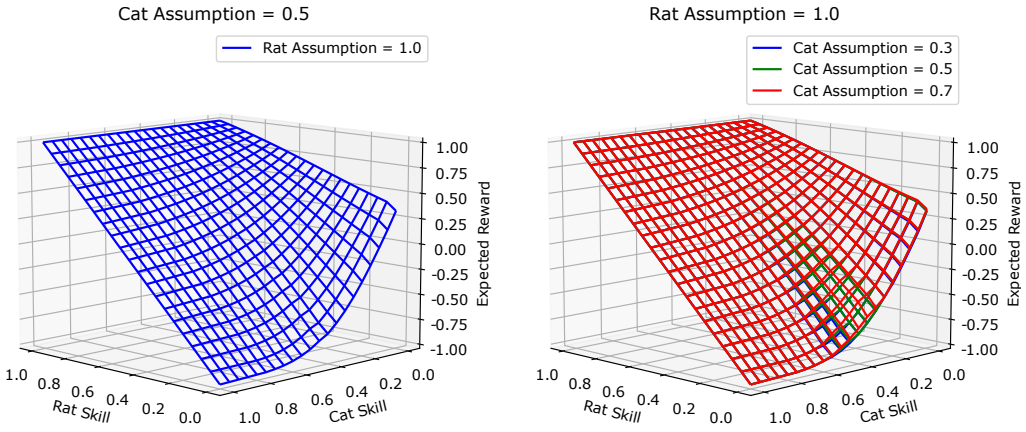


Figure 5.4: Expected reward vs player skill when $\hat{c}_{\text{cat}} = 1$ for the contrived game.

In Figure 5.5 we plot the expected reward function versus player skills for the large random game depicted on the left in Figure 4.3. For this game we notice slightly more separation between the surfaces for different assumptions. This suggests that this particular game setup induces the players' policies and resulting expected rewards to be more sensitive to the players' assumptions about their opponent's skill.

In Figure 5.6 we plot the expected reward function versus player skills for the small random game depicted on the right in Figure 4.3. Notice that this game exhibits similar features for low skilled rats and high skilled cats as we observed in the contrived game in Figure 5.2. Again this corresponds with the rat being overly optimistic that the cat cannot catch it before the end of the game when that is actually not the case.

In contrast to the contrived game, we notice some interesting behaviour in the expected reward

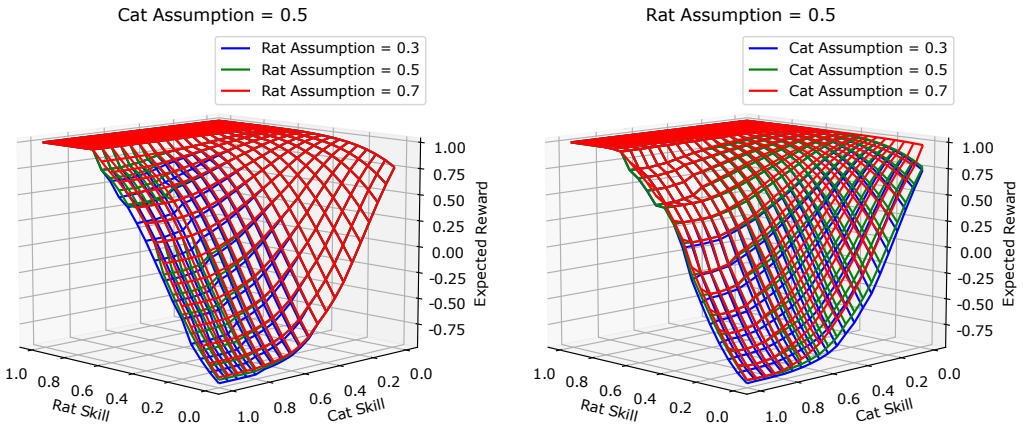


Figure 5.5: Expected reward vs player skill for the random large game.

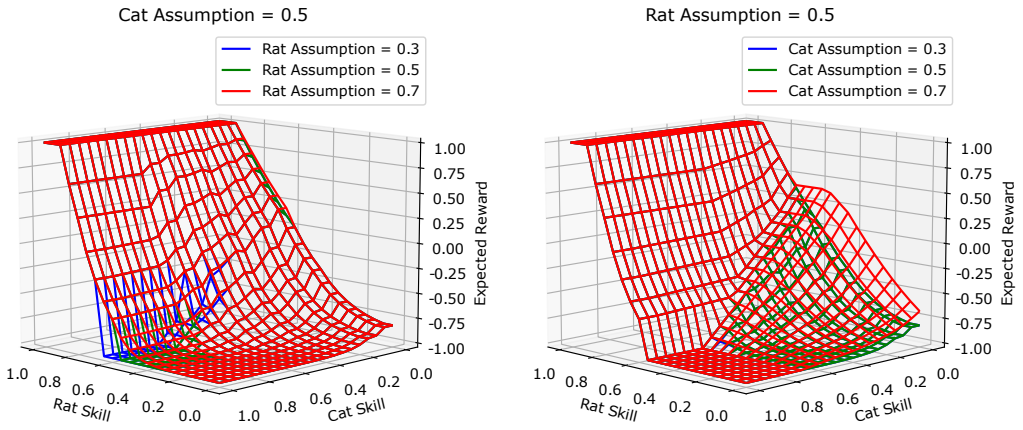


Figure 5.6: Expected reward vs player skill for the random small game.

function for rat skills in the middle of the range and cat skills close to zero. In this region the slope of the expected reward function flattens out before becoming positive again. In this particular instance, the policy that low skilled rats follow is to avoid the cat until the game ends. This strategy improves with increasing rat skill to a point and then reaches a maximum. At the point where $c_{\text{rat}} > 0.7$, the rat's optimal policy changes to trying to reach the goal since at these higher skills this strategy is more likely to result in a win for the rat.

In this section we have observed that the expected reward function is monotonically increasing along the axis of the rat's skill and monotonically decreasing along the axis of the cat's skill. We have also noted that the surfaces generated by the different player assumptions do not intersect with each other.

5.1.3 Expected Reward Versus Assumptions

Now we turn our attention to another perspective of the expected reward function for the contrived game where we choose to plot the function for varying player assumptions of their opponent's skill while holding player skills constant. In Figure 5.7 we notice that the changes in the expected reward function are a little more subtle across player assumptions than they were across skills. The separation between the different expected reward surfaces is much more pronounced in these graphs. This is due

to the change in expected reward function along the skills axes being greater than the change along the assumption axes.

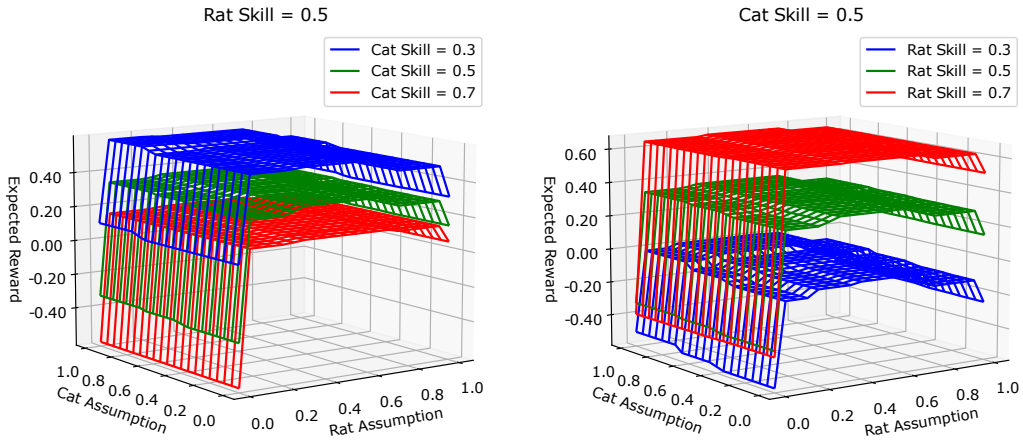


Figure 5.7: Expected reward vs player assumptions for the contrived game.

The graph on the left shows the expected reward function for rat skill $c_{\text{rat}} = 0.5$ and each surface represents the expected reward function for different cat skills $c_{\text{cat}} \in \{0.3, 0.5, 0.7\}$. The graph on the right illustrates a similar perspective with the cat skill held constant and each surface representing the expected reward function for different rat skills.

A subtle feature of these graphs of particular interest is the curvature of the expected reward function across each of the assumption axes. In order to illustrate this more clearly, we take a cross section of each graph along the points $\hat{c}_{\text{cat}} = 0.5$ and $\hat{c}_{\text{rat}} = 0.5$ of the assumption axes.

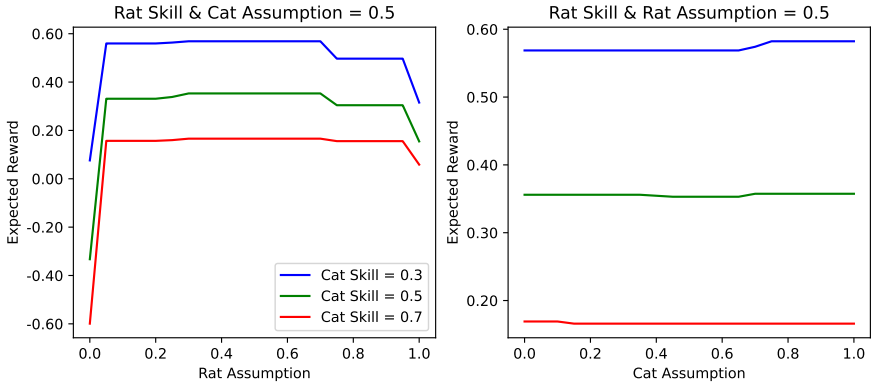


Figure 5.8: Cross section of expected reward vs player assumptions for the contrived game.

In Figure 5.8 we can see more clearly the cross section of the surfaces in the right-hand graph of Figure 5.7. In the graph on the right the rat's skill and the cat's assumption about the rat's skill match. For each curve, the rat's assumption about the cat's skill along the x-axis matches the true value of the cat's skill at different points along the axis. In the graph on the left the rat's assumption about the cat's skill matches the true value of the cat's skill for the green curve only. The cat's assumption about the rat's skill along the x-axis matches the true value of the rat's skill at 0.5.

It is now easier to observe the shape of the surface, and in particular the curvature across each assumption axis. We observe more clearly the separation between the surfaces. We also observe that along the rat assumption axis, the expected reward function is concave, with the maximum occurring

where the rat assumption matches the cat's skill. Conversely, the expected reward function is convex along the cat assumption axis, with the minimum occurring where the cat's assumption matches the rat's skill.

The curves in the graph on the right are flatter than the curves in the graph on the left in Figure 5.8. This is due to this particular game punishing the rat more harshly for holding an incorrect assumption about the cat's skill than the cat is punished for holding an incorrect assumption about the rat.

This result suggests that for any pair of player skills, the maximising player (the rat) attains the maximum possible expected reward when its assumption matches the true skill of its opponent (the cat). Similarly, the minimising player (the cat) attains the minimum possible expected reward when its assumption matches the true skill of its opponent (the rat).

The shape of the curves in Figure 5.8 provide evidence for the hypothesis that the expected reward obtained by a player making an incorrect assumption about their opponent's skill is always less than or equal to the expected reward when making the correct assumption about the opponent's skill.

We do note that due to the discrete nature of the game these local maxima and minima may occur for multiple values of the player's assumption, however the overall concavity/convexity of the expected reward function along the respective assumption axis is preserved. This means that even in this discrete environment, players suffer a loss in performance if their assumptions about their opponent's skill is sufficiently far from the truth. Furthermore, the player's performance can only get worse the further their assumption is from the truth. Players do not derive any benefit in performance from having incorrect assumptions about their opponent's skill.

While we do not have a formal proof of this result, we present experimental evidence that the structure of the expected reward function is concave along the rat assumption axis, and convex along the cat assumption axis. In Figures 5.9 and 5.10 that follow we present the equivalent plots to Figure 5.7 for the randomly generated large and small games respectively.

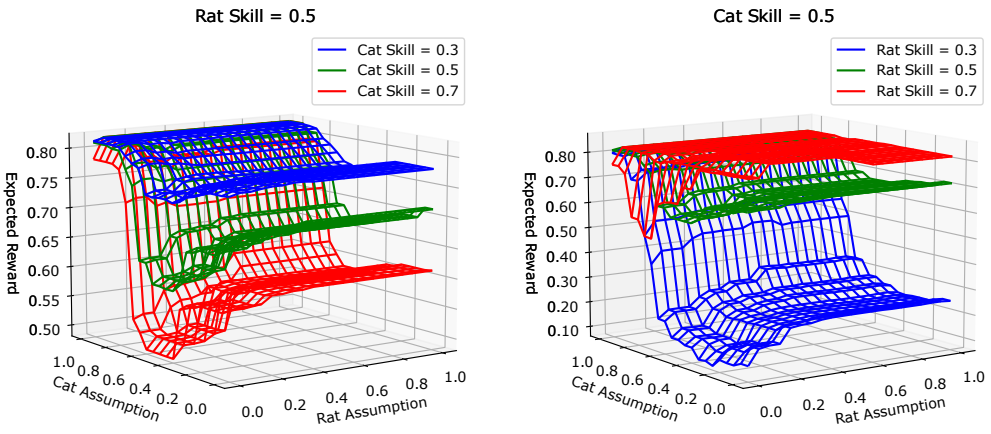


Figure 5.9: Expected reward vs player assumptions for the random large game.

Again we notice the concavity of the expected reward function along the rat assumption axis, and the convexity along the cat assumption axis.

In this section we have observed that the expected reward function is monotonically increasing along the skill axis of the maximising player (rat) while being monotonically decreasing along the skill

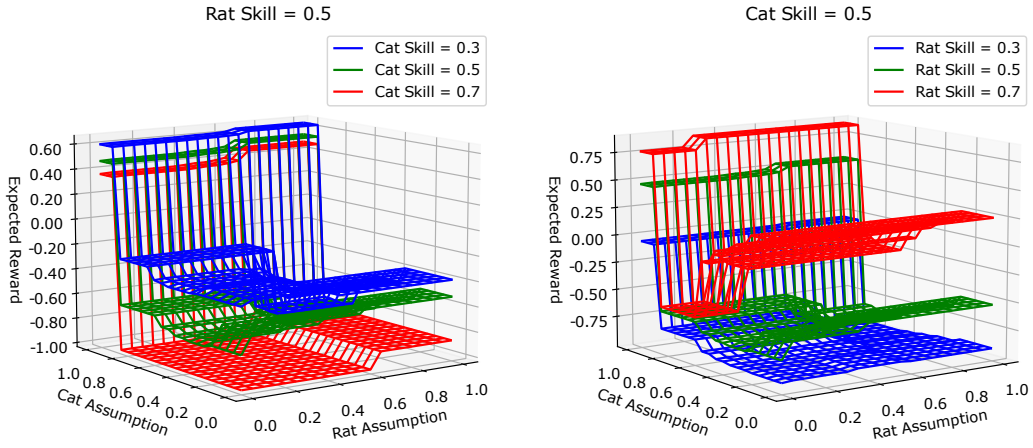


Figure 5.10: Expected reward vs player assumptions for the random small game.

axis of the minimising player (cat).

Along the assumption axes we observed that the expected reward function is concave along the rat's assumption axis and convex along the cat's assumption axis. We note that the rat attains the maximum possible expected reward where its assumption matches the true skill of the cat, and the cat attains the minimum possible expected reward when its assumption matches the true skill of the rat. Due to the discrete nature of the game environment, these local maxima and minima along the assumption axes may occur for multiple discrete values of player assumptions. This provides evidence in support of our hypothesis that the expected reward obtained by a player making an incorrect assumption about their opponent's skill is always less than or equal to the expected reward when making the correct assumption about the opponent's skill.

We have observed that the increase in expected reward is typically greater for higher player skill than it is for a more accurate assumption about the opponent's skill. This raises an interesting observation in this game environment that if a player had a choice between higher skill or a more accurate assumption it should choose the higher skill as this would typically lead to the larger performance increase.

5.2 Effects of the Failure Factor

In this section we consider the effect of the failure factor, κ, on the expected reward function. Recall that the failure factor shifts the transition function in Equation 4.1.1 downwards for increasing values of κ. In the previous section we chose $\kappa = 0$ and we now consider the case where $\kappa = 0.01$. This has the effect of increasing the chance of action failure where players would otherwise have been able to transition with certainty across edges with weights less than or equal to their skill.

In comparing Figure 5.11 to Figure 5.2 the most noticeable difference is the scale of the z-axis. For the case where $\kappa = 0$ the range of the expected reward function was $[-1, 1]$. Figure 5.12 is a heat-map of the differences between the top surfaces in Figure 5.2 and Figure 5.11. This heat-map better illustrates the change in scale of the z-axis. In the case where $\kappa = 0.01$ we notice that the range of the z-axis is approximately $[-0.8, 0.8]$. This makes intuitive sense as the additional transition uncertainty introduced by setting $\kappa > 0$ results in the expected reward being smaller in magnitude for both players.

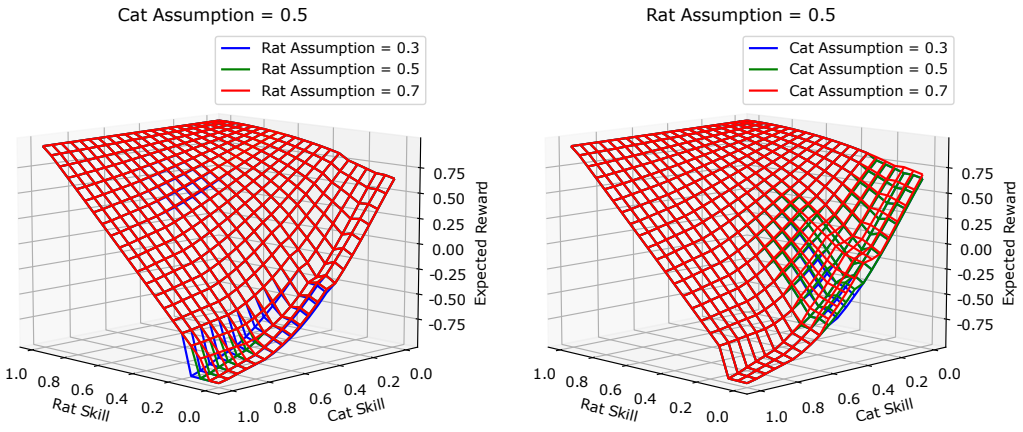


Figure 5.11: Expected reward vs player skill for the contrived game with $\kappa = 0.01$.

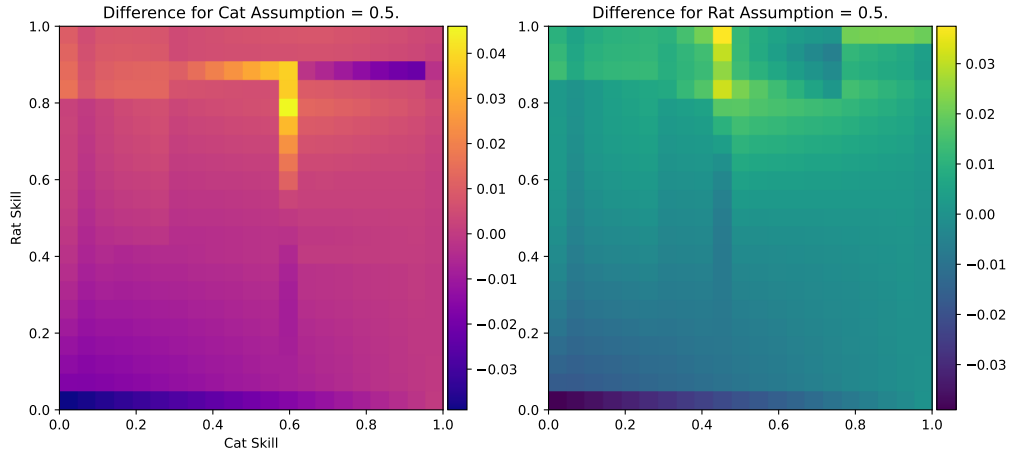


Figure 5.12: Heat-map of the differences between the top level surfaces of Figure 5.11 and Figure 5.2.

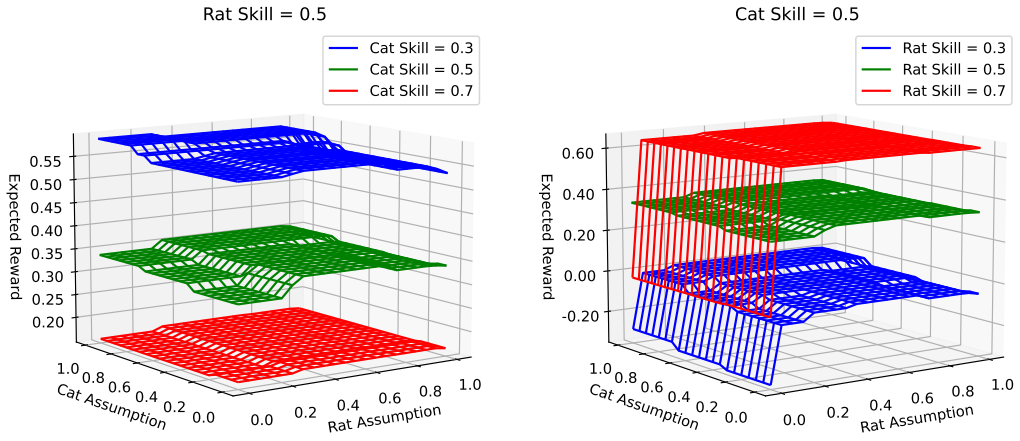


Figure 5.13: Expected reward vs player assumptions for the contrived game with $\kappa = 0.01$.

Comparing Figure 5.13 to Figure 5.7, the most striking difference is along the edges of the two graphs where the rat assumption of the cat's skill is zero or one, $\hat{c}_{\text{cat}} = \{0, 1\}$. In the graph on the left we notice that the sharp drop in the expected reward function that was previously at the extreme edges of the surface along the rat assumption axis has now disappeared. In the graph on the right,

the similar sharp drop has been reduced where $\hat{c}_{\text{cat}} = 0$ and had disappeared where $\hat{c}_{\text{cat}} = 1$.

This suggests that the increased uncertainty has benefited the rat in this game when the rat assumes the cat has skill of either 0 or 1 when in fact the cat's skill is in the middle of the range $0.3 \leq c_{\text{cat}} \leq 0.7$. Here the uncertainty benefits the rat when making extreme assumptions about the cat's skill as it needs to factor in the additional uncertainty in the transition dynamics for both itself and its opponent. This has the effect of encouraging the rat to play more conservatively than it would under either assumption about the cat's skill when $\kappa = 0$.

When $\kappa = 0$, the rat is confidently wrong in its extreme assumptions about the cat's skill and the game punishes the rat accordingly. When $\kappa > 0$, the rat no longer places as much confidence in its assumption of the cat's skill when determining the transition dynamics. As such, the rat plays more cautiously when $\kappa > 0$ which has the effect of benefiting the rat when its assumptions are wrong in the extreme.

Now we observe the same graphs in Figure 5.14 for the expected reward function of the large randomly generated game for $\kappa = 0.01$.

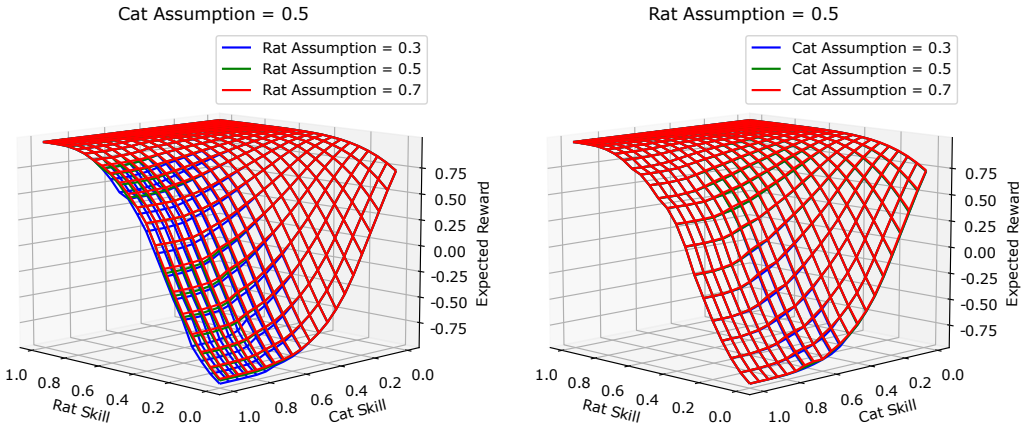


Figure 5.14: Expected reward vs player skill for the random large game with $\kappa = 0.01$.

When comparing Figure 5.14 with Figure 5.5 we do notice some subtle changes in the separation between the surfaces in both graphs. Similarly to the contrived game, the expected reward range is reduced in magnitude due to the increased uncertainty in the transition dynamics.

Comparing Figure 5.15 with Figure 5.9 we again notice some subtle changes on the extreme edge of the rat assumption axis where the rat's assumption of the cat's skill is zero, $\hat{c}_{\text{cat}} = 0$. We also notice that the cat is now punished less for being confidently wrong where it assumes $\hat{c}_{\text{rat}} = 1$.

Under $\kappa = 0$, for the blue surface in the left hand graph (Figure 5.9) the cat performs poorly when its assumption about the rat's skill approaches 1. The cat performs marginally better when $\kappa = 0.01$ (Figure 5.15) due to the increased uncertainty in the transition dynamics resulting in a more cautious policy from the cat.

Finally for this section we observe the same graphs for the expected reward function of the small randomly generated game for $\kappa = 0.01$. When comparing Figure 5.16 with Figure 5.6 we observe again that the expected reward range is reduced in magnitude due to the increased uncertainty in the transition dynamics.

Comparing Figure 5.17 with Figure 5.10 we observe an improvement in performance for the cat where it assumes the rat's skill is $\hat{c}_{\text{rat}} = 1$. Similar to the randomly generated large game, the cat is

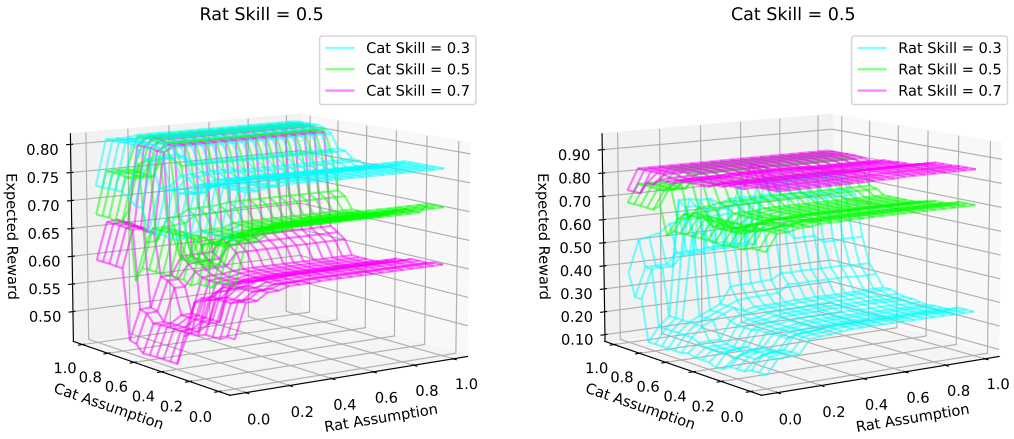


Figure 5.15: Expected reward vs player assumptions for the random large game with $\kappa = 0.01$.

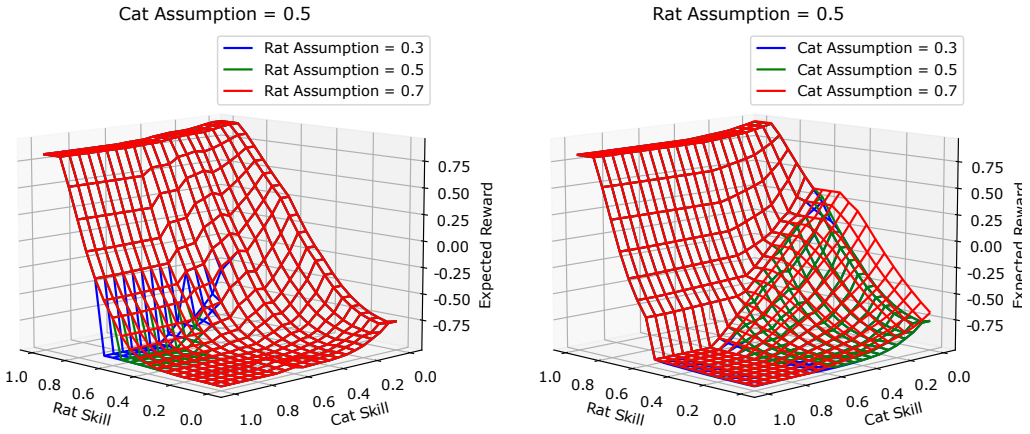


Figure 5.16: Expected reward vs player skill for the random small game with $\kappa = 0.01$.

punished for being confidently wrong in the extreme about the rat's skill, but when $\kappa > 0$, this effect appears to have disappeared.

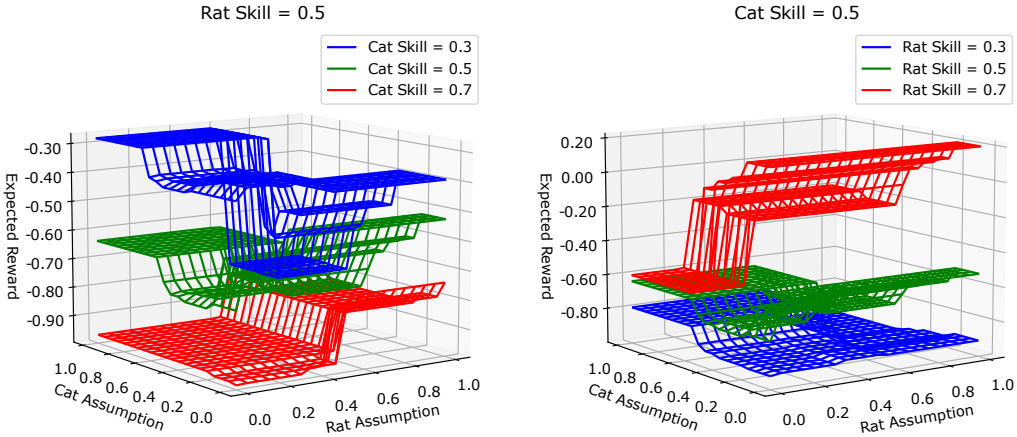


Figure 5.17: Expected reward vs player assumptions for the random small game with $\kappa = 0.01$.

In this section we have observed the effects of a positive failure factor, κ, through a series of experiments. The first effect is that the overall magnitude of expected rewards is reduced for both

players. The second effect is that the increased failure factor can benefit some players by encouraging them to play more conservatively. This results in players suffering less from the consequences of an assumption about the opponent's skill that is extremely wrong.

5.3 Effects of Discounting the Expected Reward

In this section we consider the effect of the discounting parameter γ in producing the discounted expected reward function. In Section 5.1.1 we chose $\gamma = 1$ and we now consider the case where $\gamma = 0.99$. In order not to conflate the effects of discounting with the effects of the failure factor, we choose $\kappa = 0$. This way the discounted expected rewards in this section can be compared to the expected rewards in Section 5.1.1 to ascertain the effects of discounting in isolation.

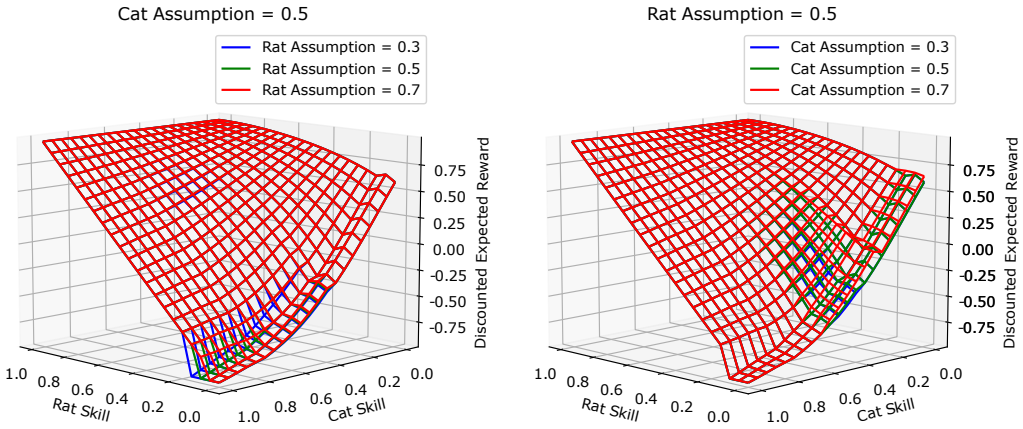


Figure 5.18: Discounted expected reward vs player skill for the contrived game with $\gamma = 0.99$.

In comparing Figure 5.18 to Figure 5.2 the only apparent difference is the scale of the z-axis. For the case where $\gamma = 1$ the range of the expected reward function was $[-1, 1]$. In the case where $\gamma = 0.99$ we notice that the range of the z-axis is approximately $[-0.8, 0.8]$. This makes intuitive sense as we are now computing the discounted expected reward at $t = 0$.

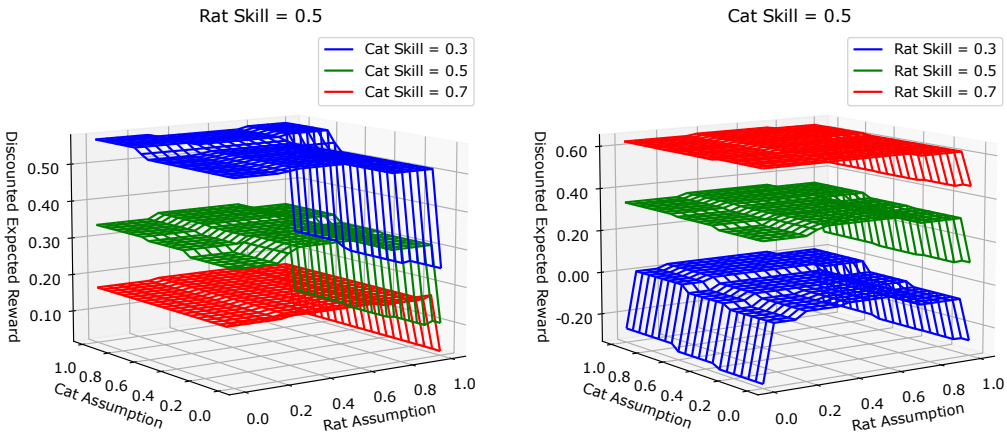


Figure 5.19: Discounted expected reward vs player assumptions for the contrived game with $\gamma = 0.99$.

Comparing Figure 5.19 to Figure 5.7, we notice for the blue surface on the left hand graph, the discounted expected reward function is concave where the cat assumption is in the range $0.8 \leq \hat{c}_{\text{rat}} \leq 1$.

At first glance this may seem alarming as we expect the reward function to be concave along the rat assumption axis.

Upon further investigation it was determined that the discounting introduces an undesirable trait in certain games. This trait occurs when, due to discounting the reward, some players (in this case the rat) may end up choosing a shorter path to a more uncertain reward over a longer path to a more certain reward. Since we are now multiplying the expected reward by γ at each time step, this reduces the value of rewards received later in the game. Since rewards are also multiplied by the transition probabilities when computing the expected reward, it impossible to disentangle the discounting of the reward with the probability of receiving that reward.

Consider a case where $\gamma = 0.5$ and a player has the choice between receiving a certain reward of 1 after two actions along a given path, or receiving an uncertain reward of 1 with probability 0.5 after one action along a different path. The discounted expected reward of the first choice is 0.5 as is the discounted expected reward of the second choice. The rat is unable to distinguish between the reward being reduced by discounting and the reward being reduced by the transition probability.

This experiment demonstrates that in this kind of Markov game environment, introducing discounting into the computation of the expected reward function may have undesirable effects on the players' optimal policies. Caution should be taken when introducing discounting, and the decision should be justified.

The use of discounting could be appropriate where players are rewarded for winning in the fewest moves possible. The choice of discount factor γ appears to determine how risk-seeking the players are although the extent of this impact remains unexplored.

Now we observe the same graphs for the expected reward function of the large randomly generated game for $\gamma = 0.99$. When comparing Figure 5.20 with Figure 5.5 we do notice some subtle changes in the surfaces in both graphs. Similarly to the contrived game, the discounted expected reward range is reduced in magnitude due to discounting.

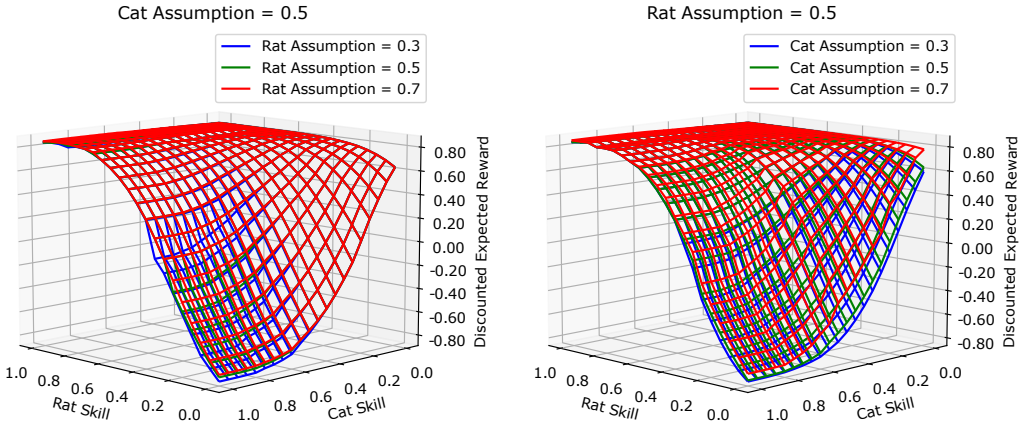


Figure 5.20: Discounted expected reward vs player skill for the random large game with $\gamma = 0.99$.

Comparing Figure 5.21 with Figure 5.9 we notice some subtle changes on the extreme edge of the rat assumption axis where the rat's assumption of the cat's skill is zero, $\hat{c}_{\text{cat}} = 0$. It appears the discounting has reduced the punishment the rat receives for being confidently wrong.

Finally for this section we observe the same graphs for the expected reward function of the small randomly generated game for $\gamma = 0.99$. When comparing Figure 5.22 with Figure 5.6 we observe again

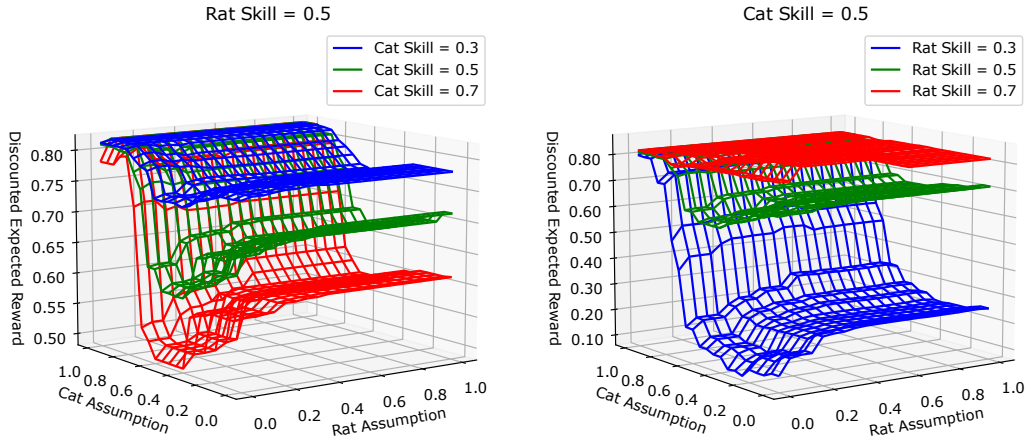


Figure 5.21: Discounted expected reward vs player assumptions for the random large game with $\gamma = 0.99$.

that the expected reward range is reduced in magnitude due to discounting the expected reward. A heat-map of the differences between the surfaces of Figure 5.6 is provided in Figure 5.23 for clearer visualisation.

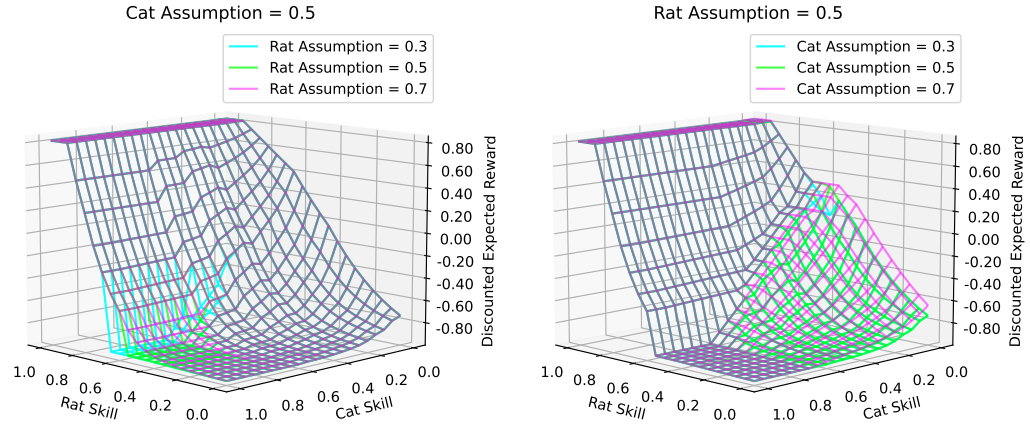


Figure 5.22: Discounted expected reward vs player skill for the random small game with $\gamma = 0.99$.

Comparing Figure 5.24 to Figure 5.10 we notice that the punishment for the cat being extremely pessimistic in its assumption of the rat's skill ($\hat{c}_{\text{rat}} = 1$) is not punished as severely with the discounting as it is ordinarily.

In this section we have observed that choosing $\gamma < 1$ can have some interesting and sometimes undesirable effects on the expected reward function and player behaviour. When players try to maximise/minimise the discounted expected reward, it may, in some cases, encourage those players to become risk-seeking. These risk-seeking players prefer a less likely reward over a more likely reward if it takes fewer moves to receive the less likely reward. Choosing $\gamma < 1$ encourages risk-seeking behaviour in the players, which undermines the effects of the players' assumptions about their opponent's skill on the expected reward function.

Since this research report is primarily focused on player skills and how best to estimate them, it would not be sensible to use discounting in our estimation analysis. For the remainder of this document, we choose $\gamma = 1$ and rather consider other factors when estimating skills.

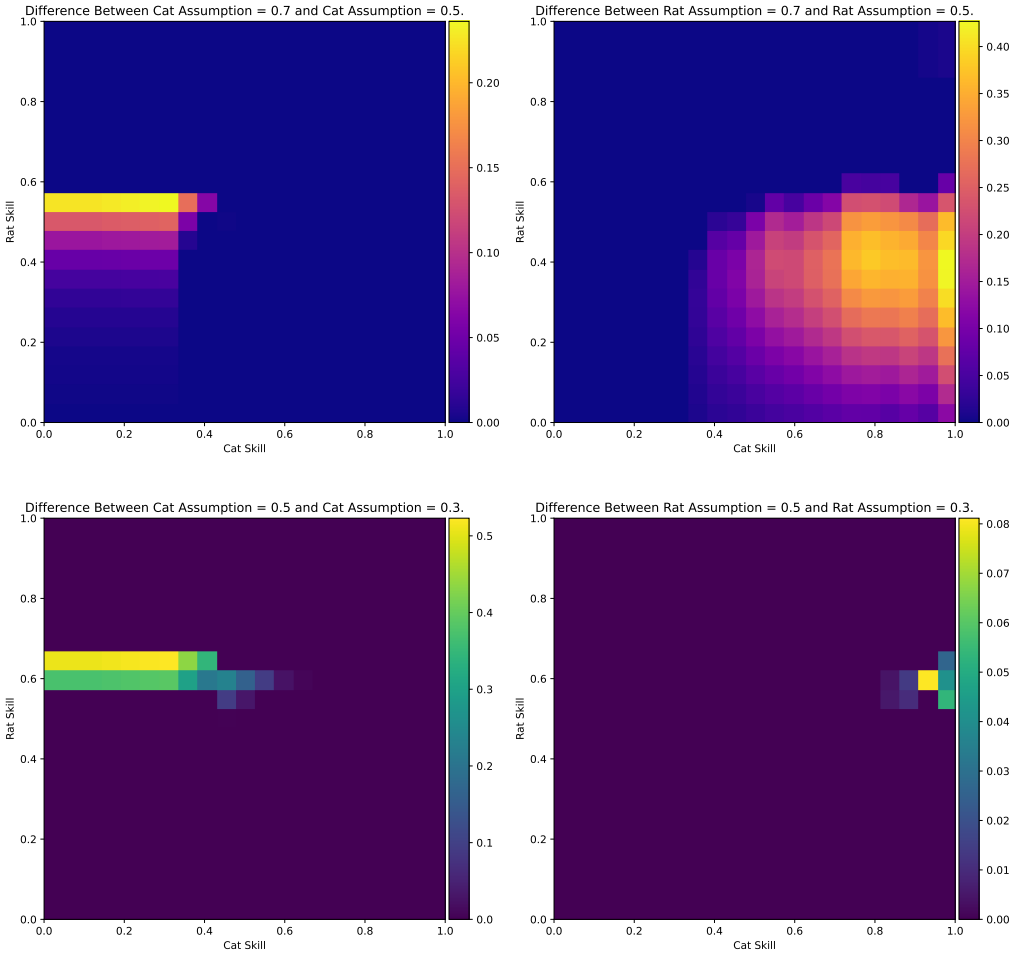


Figure 5.23: Heat-map of the differences between the surfaces of Figure 5.22.

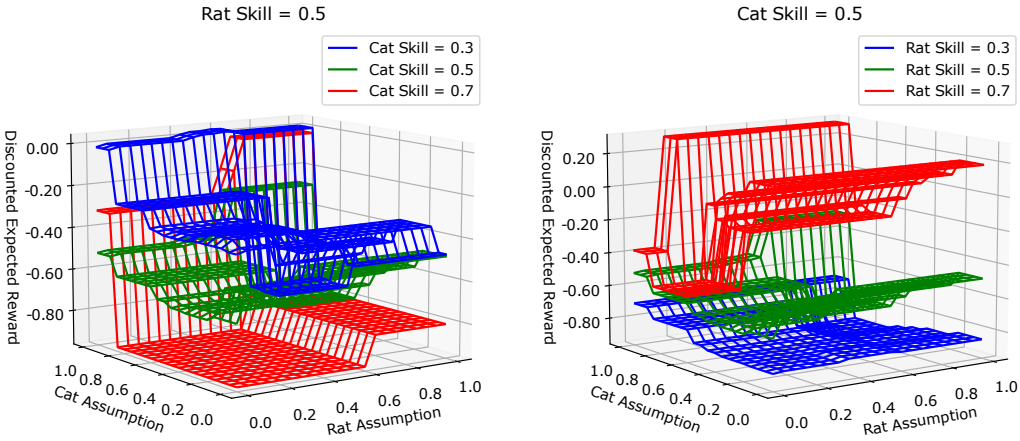


Figure 5.24: Discounted expected reward vs player assumptions for the random small game with $\gamma = 0.99$.

5.4 Effects of a Different Reward for Ties

Recall that in Equation 4.2 we had a reward β for players when the game ends in a tie. In this section we choose $\beta = 0$ which has the effect of giving no reward to players when the game reaches

the terminal time $t = T$. It is important to understand that a score of zero may still be considered the optimal reward by a player in situations where any alternative play results in the opponent winning. Intuitively, when a player is outmatched in skill with no hope of winning, playing for a draw may be the best they can achieve.

Observing the expected reward function for this version of the game in Figure 5.25 and comparing it to the expected reward function illustrated in Figure 5.2 we notice only minor differences in the expected reward functions for the contrived game. It appears that the small amount of separation between the surfaces in the graph on the left has become less pronounced, while the separation between the surfaces in the graph on the right has become slightly more pronounced.

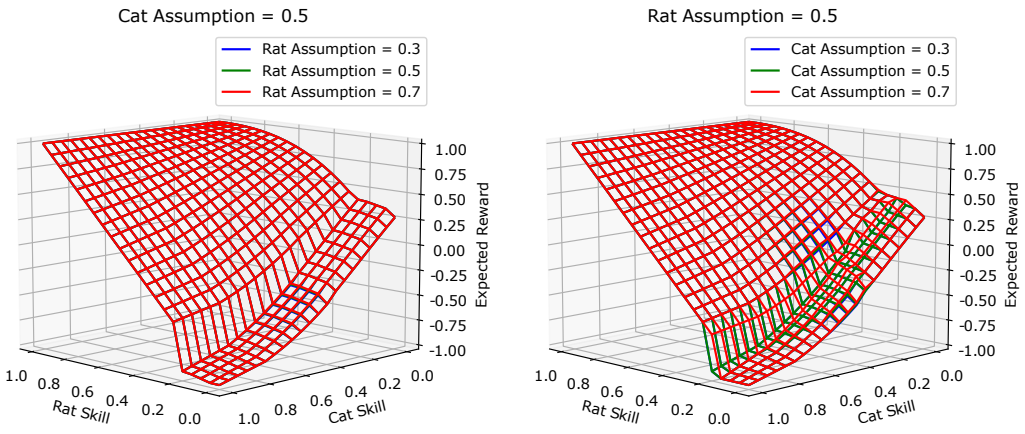


Figure 5.25: Expected reward vs player skills for the contrived game with modified rewards.

Comparing the expected reward function in Figure 5.26 to that in Figure 5.7 we again notice only minor differences. The expected reward function along the rat assumption axis seems to have become slightly less concave under the alternative scoring.

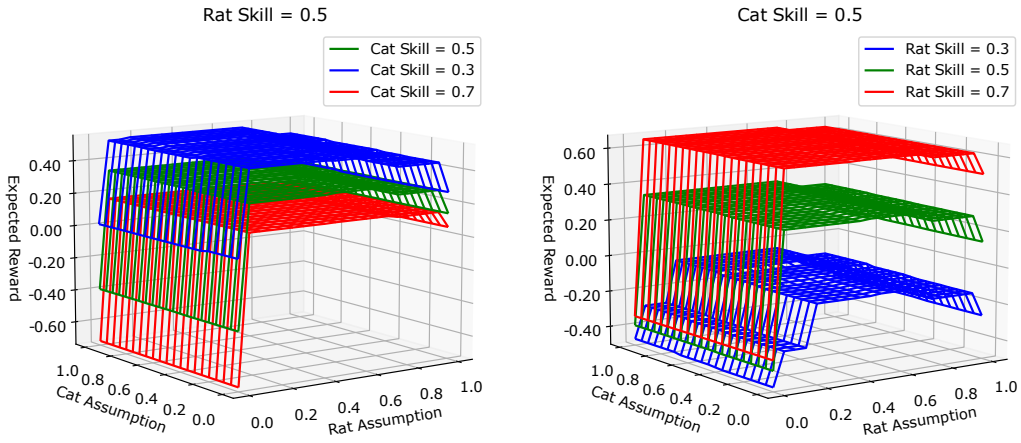


Figure 5.26: Expected reward vs player assumptions for the contrived game with modified rewards.

When observing the expected reward function for the large randomly generated game in Figure 5.27 under the alternative scoring we notice that the surfaces have changed significantly from those observed in Figure 5.2. The introduction of the draw has reduced the range of the expected reward function from $[-1, 1]$ to approximately $[-0.6, 0.8]$. This suggests that for this game instance, playing for the draw and getting a score of zero is an outcome that is often more easily attainable by both

players than a win. The alternative scoring setup appears to encourage the players in this instance of the game to play more conservatively by choosing to draw the game more often.

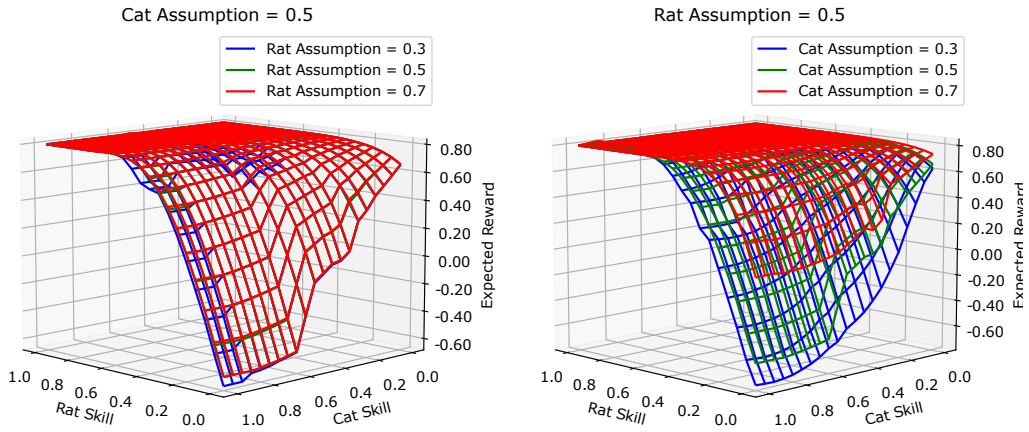


Figure 5.27: Expected reward vs player skills for the random large game with modified rewards.

The most striking changes in the expected reward function thus far occur in Figure 5.28. Comparing these graphs to those in Figure 5.7 the shape of the surfaces have changed almost completely. The structure of the expected reward surfaces are still concave along the rat assumption axis and convex along the cat assumption axis.

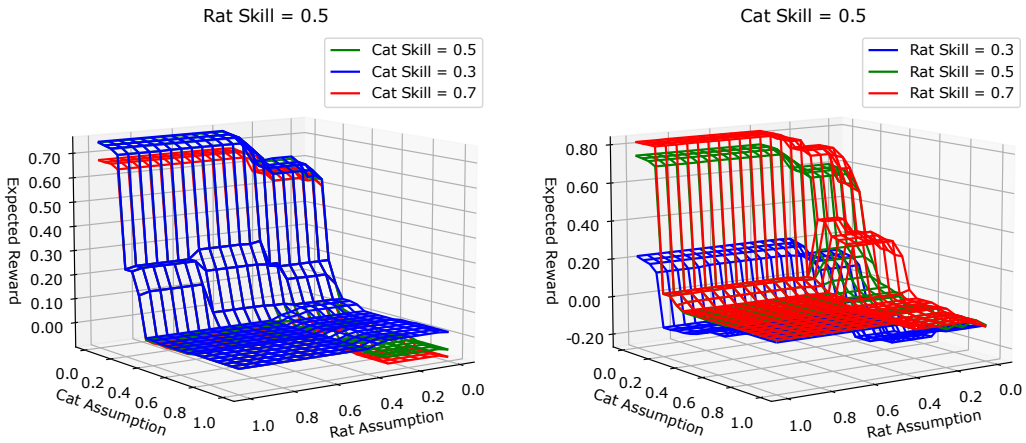


Figure 5.28: Expected reward vs player assumptions for the random large game with modified rewards.

Finally we consider the randomly generated small game under the alternative reward scoring. In this game, it is always too risky for the rat to go for the goal as the rat almost always gets blocked or caught by the cat. Under either scoring regime, the rat's optimal strategy is to evade the cat until the end of the game. Under the initial scoring regime, the rat would still get a score of 1, but under the alternative regime the rat gets a score of 0.

We observe this in Figure 5.29 where the expected reward function is now in the range $[-1, 0]$. The overall shape and separation remains similar to the surfaces in Figure 5.6. There are some differences in the shape of the surface where the cat's skill is low and the rat skill is around 0.5.

The expected reward function surfaces in Figure 5.30 have changed compared to the surfaces in Figure 5.10, particularly where the cat assumption about the rat's skill is close to 1. This makes sense since the rat's performance is capped at zero by attaining a draw. Thus the cat can perform better

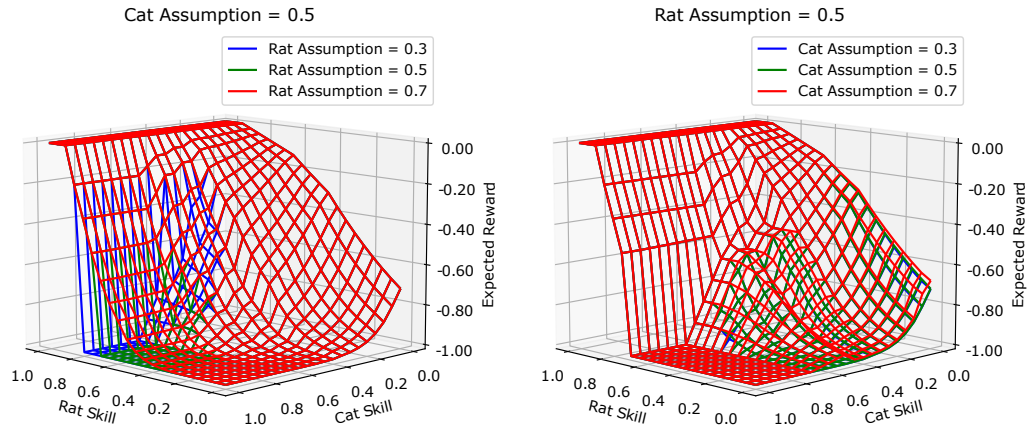


Figure 5.29: Expected reward vs player skills for the random small game with modified rewards.

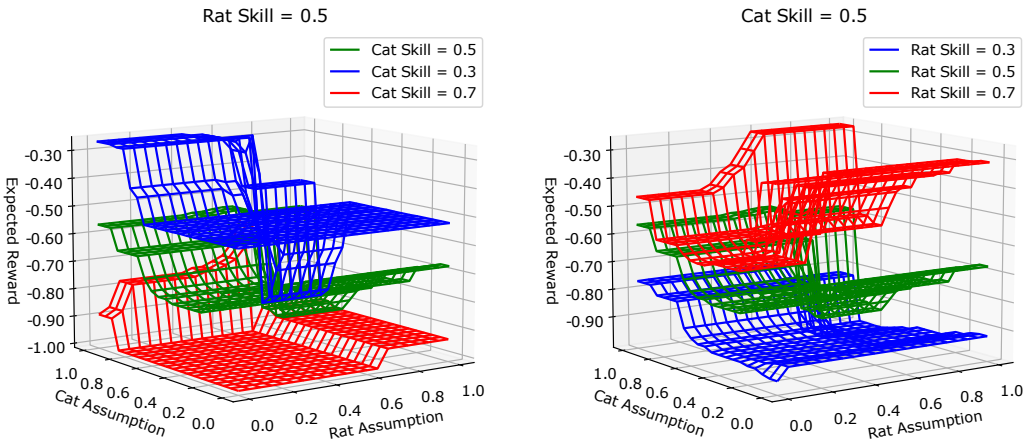


Figure 5.30: Expected reward vs player assumptions for the random small game with modified rewards.

knowing the rat never wins regardless of its skill.

In this section we have observed the effects of choosing $\beta = 0$ as an alternative scoring system for the game compared to the preceding sections. In some instances of the game we noticed that this choice has the effect of making some players more conservative. It can also have the effect of changing the shape of the expected reward function along certain axes significantly.

Below we provide an alternative arrangement of some of the graphs from above so they can be viewed side-by-side for ease of comparison. In Figure 5.31 we provide a side-by-side comparison of the expected reward graphs for the cat playing the contrived game under the base experiment, the positive failure factor experiment, the discounting experiment, and the alternative reward experiment.

Here we can more clearly observe the effects of the different experiments on the expected reward function of the cat.

In Figure 5.32 we provide a side-by-side comparison of the expected reward graphs for the cat playing the contrived game under the base experiment, the positive failure factor experiment, the discounting experiment, and the alternative reward experiment.

In this chapter we observed in Section 5.1.2 that the expected reward function is monotonically increasing along the axis of the rat's skill and monotonically decreasing along the axis of the cat's skill. We have also noted that the surfaces generated by the different player assumptions do not intersect

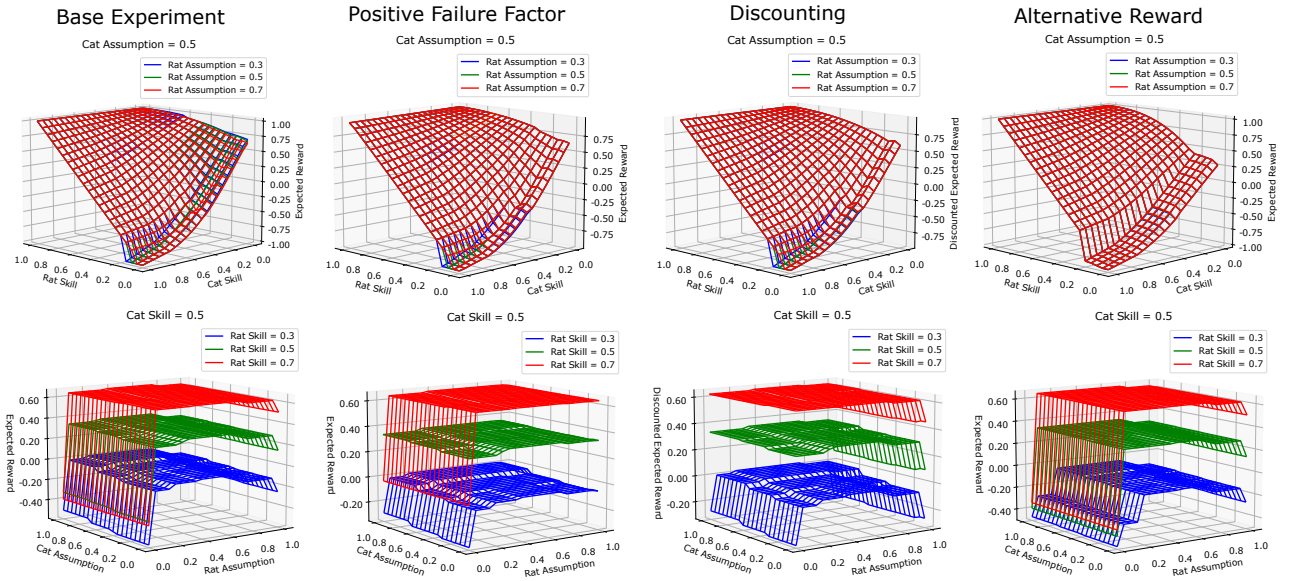


Figure 5.31: Side-by-side comparison of cat graphs for the contrived game.

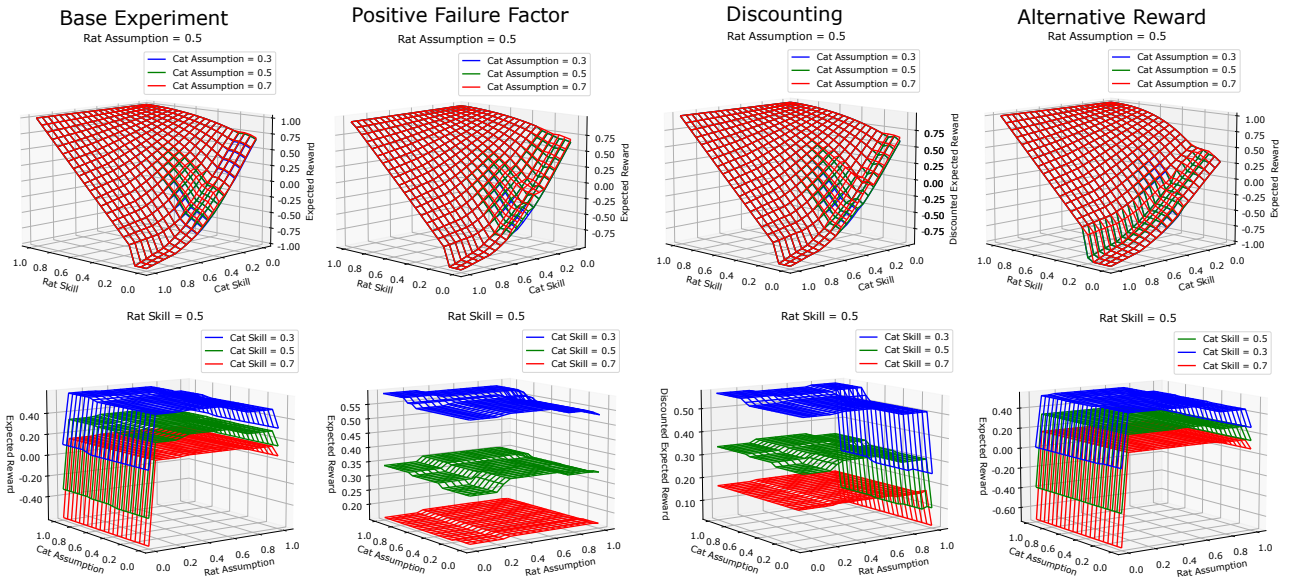


Figure 5.32: Side-by-side comparison of rat graphs for the contrived game.

with each other.

In Section 5.1.3 we observed that the expected reward function is concave along the rat's assumption axis and convex along the cat's assumption axis. We note that the rat attains the maximum possible expected reward where its assumption matches the true skill of the cat, and the cat attains the minimum possible expected reward when its assumption matches the true skill of the rat.

From the results presented in Sections 5.1.2 and 5.1.3 we note that, in this game environment, if a player had a choice between higher skill or a more accurate assumption it should choose the higher skill as this would typically lead to the larger performance increase.

In Section 5.2 we observed that a positive failure factor, $\kappa > 0$, has the effect of reducing the overall magnitude of expected rewards for both players. The increased failure factor can also benefit some players by encouraging them to play more conservatively. This results in players suffering less

from the consequences of an assumption about the opponent's skill that is extremely wrong.

In Section 5.3 we observed that choosing $\gamma < 1$ may, in some cases, encourage players to become risk-seeking. These risk-seeking players may choose a less likely reward over a more likely reward if it takes fewer moves to receive the less likely reward. The use of discounting can make it impossible for players to differentiate between receiving a reward later with certainty and receiving a reward sooner with uncertainty.

Finally in Section 5.4 we observed that choosing $\beta = 0$ has the effect of making some players more conservative in some games and changes the shape of the expected reward function along certain axes significantly.

Chapter 6

Skill Estimation

In this chapter we conduct experiments that demonstrate that players can estimate the skill of their opponent and that they can use this estimate to improve their performance when playing. We explore how players who choose their actions randomly rather than using Expectiminimax, can impact the skill estimation done by players who assume their opponent is using Expectiminimax. Recall that when estimating the opponent's skill, the likelihood function in Equation 4.9 is a function of the cat's optimal policy π^*_{cat} as computed by the Expectiminimax algorithm. When a cat behaves randomly, they are no longer following the optimal policy π^*_{cat}. By conducting the skill estimation experiment on random players we test how robust our methodology is when estimating the skills of sub-optimal players.

In Section 6.1 we conduct our the first experiment in which players make use of the likelihood function described in Equation 4.9 to update a prior joint distribution of the opponent's skill and assumption. The prior is updated based on the moves made by the opponent in one simulated game. We observe that in our discrete pursuit-evasion Markov game, that this joint distribution of opponent skill and assumption does converge to the unseen (by the player) parameter values in some cases. We also observe that in other cases, the discrete setting may not provide enough variety of play to disambiguate between a range of skill and assumption parameters no matter how much play is observed.

In Section 6.1, once the posterior distribution of the opponent's skill and assumption is determined, we conduct our second experiment in which players use the posterior distribution to inform their own mixed optimal policy and, in doing so, can improve their expected performance. As with the first experiment, we observe cases where the skill estimate converges quickly and player performance is rapidly improved to near optimal levels. We also observe cases where the estimate converges slowly or not at all, resulting in players performing at least as well compared to not using skill estimation.

In Section 6.2 we conduct our final experiment for this chapter. In this experiment we pit players against a random-action opponent that does not use the Expectiminimax algorithm and instead selects actions at random during the simulated games. We observe that in some instances this can cause the skill estimation of a player facing a random-action opponent to converge rapidly and, in other instances, may cause the skill estimation to perform poorly. In these instances, players using estimation perform at least as well as compared to not using skill estimation.

For all three experiments we choose the player skill to be $c_{\text{rat}} := 0.5$ as this is a middle of the range skill level which requires the player to reason about opponents that have more skill and opponents that have less skill. We choose the opponent's skill to be $c_{\text{cat}} := 0.4$ as this provides the player with

a slight skill advantage over the opponent. By making this choice we expect to see the player behave more conservatively at first and, once it has localised the opponent's skill, to exploit its skill advantage over the opponent. We choose the opponent assumption to be $\hat{c}_{\text{rat}} := 0.5$ which matches the players skill. This gives the opponent an advantage in the beginning of the estimation problem as it has been given perfect foresight of the player's skill.

We consider the effects of the failure factor taking on values $\kappa = 0, 0.01$ as this parameter directly affects the transition probabilities that are used to compute the likelihood function. In this chapter we also choose $\beta = 1$ and $\gamma = 1$ to create results consistent with those presented in Sections 5.1 and 5.2.

Consider the rat computing their optimal policy $\pi^*_{\text{rat}}(\hat{c}_{\text{cat}})$ for discrete values of $\hat{c}_{\text{cat}}$ that span the skill range $[0, 1]$. Let $\hat{c}^i_{\text{cat}}$ for $i = 1, ..., n$ be this discretisation of the rat's assumptions about the cat's skill. The rat can construct a mixed optimal policy strategy by selecting the discrete optimal policy $\pi^*_{\text{rat}}(\hat{c}^i_{\text{cat}})$ for each assumption $\hat{c}^i_{\text{cat}}$ in proportion with the probability that the rat believes $\hat{c}^i_{\text{cat}} = c_{\text{cat}}$ obtained from the posterior distribution. We assume that players use a mixed optimal policy strategy when using the posterior distribution to play. Since the opponent's assumption does not affect the player's computation of the expected reward function, we can use the marginal distribution of opponent skill when computing the mixed strategy. Players compute the distribution for a discrete set of opponent skills, and can then compute the expected reward function for each point as in Figure 5.8.

Multiplying the marginal probabilities by the corresponding expected reward for that assumption of opponent skill and summing the results we compute the expected reward from a mixed optimal policy where the player assumes a given opponent skill in proportion to its marginal probability. We can then use this metric to compare the performance of players that use estimation with those that do not.

In this chapter we restrict our analysis to the contrived game depicted in Figure 4.2. Similar analysis has been conducted for the games depicted in Figure 4.3 but the results from these games did not contribute any additional observations beyond those seen in the contrived game. The interested reader can find the respective posterior distribution and estimating player performance graphs for these games in Appendix A.

6.1 Skill Estimation of Expectiminimax Players

In this section we consider players that use the Expectiminimax optimal policy and estimate the skill of the opponent based on observed play. We then compute the expected reward players receive when using their posterior distribution of the opponent's skill to inform a mixed optimal policy strategy.

Each player begins with a uniform prior distribution across skill and assumption of the opponent. In the plots that follow we present the state of the estimation distribution (estimator) after a given number of observed games.

The estimation algorithm has been set up to terminate after a maximum of 10 simulated games or when player performance is within a certain threshold of optimal performance. For this optimal performance metric we consider a player with perfect knowledge of the opponent's skill which we call the Oracle. Since the Oracle knows its opponent's skill it can simply use this value in the place of any assumption and achieve its optimal expected reward as computed in the preceding chapter. As the

simulated games are dependent on random transition probabilities, the outcomes of these games are seed dependent.

We begin our first experiment with the cat trying to estimate the rat's skill and assumption about the cat's skill in our contrived game with $\kappa = 0$.

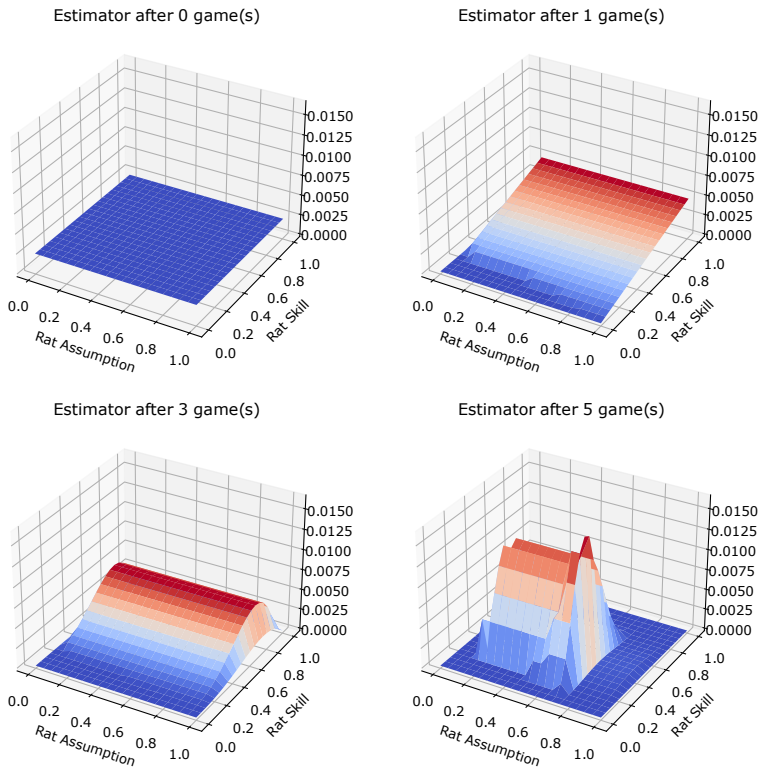


Figure 6.1: Posterior distribution of cat estimating rat in the contrived game.

In Figure 6.1 we see at the top left we start with the joint uniform prior distribution across the rat's skill and assumption about the cat's skill. In the top right graph we observe the state of the distribution after one game has been played. We can see that at this point, the rat having a skill of $c_{\text{rat}} = 0$ has been ruled out by the observed play.

After 3 games, in the bottom left graph, we observe that the estimator is now more peaked, placing less probability on the extreme edges of the skill axis. Up until this point, there has been no observed play to disambiguate the rat's assumption about the cat's skill. As we observed in the previous chapter, it is often the case that the player assumptions have a smaller effect on the expected value function and player policies than player skills have.

After 5 games, in the bottom right graph, we observe that the distribution has become far more peaked along the rat skill axis, with the peak coinciding with the true value of the rat's skill. We also notice that further play has allowed the estimator to rule out some values for the rat's assumption.

The results in Figure 6.1 provide evidence in support of our hypothesis that when players use Bayesian inference in estimating their opponent's skill, their posterior distribution concentrates probability mass around the true value of the opponent's skill when sufficient play is observed.

Using the prior distribution and posterior distributions after each simulated game, we plot the trajectory of the cat's expected reward when using these distributions to create a mixed optimal policy. In the same plot we also include the slice of the expected reward function which pertains to the simulated games where the cat uses static assumptions about the rat's skill. We also mark the

point which coincides with the Oracle's performance.

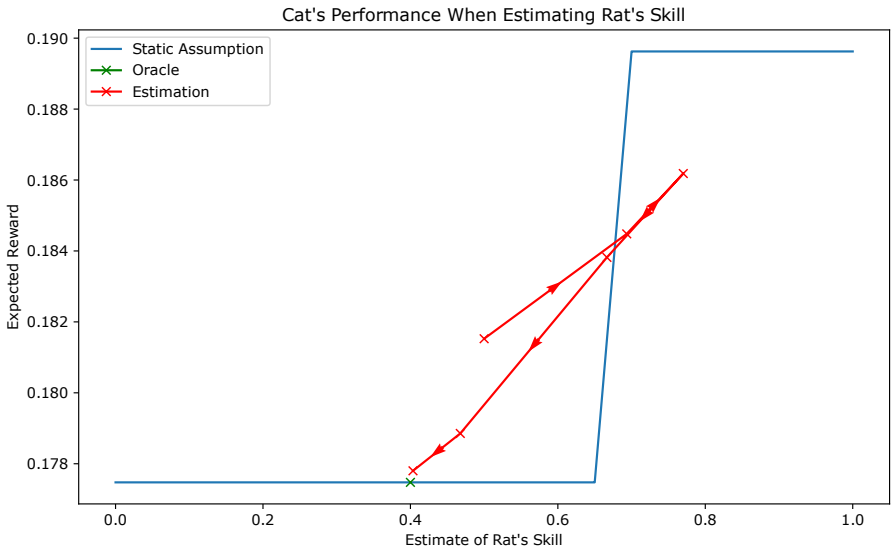


Figure 6.2: Expected reward of cat estimating rat in the contrived game.

In Figure 6.2 we first observe that the blue static assumption curve is indeed convex. This curve is comparable with those in Figure 5.8 and is a slice of the expected reward function in Figure 5.7. The fact that this curve is convex means there is a minimum expected reward (or rewards) for which the cat, as the minimising player, is striving to achieve. We also notice that the Oracle point is at one point of minimum expected reward. We notice that, in this discrete game, there may be multiple values for the cat's assumption about the rat's skill for which the optimal expected reward may be achieved. In this game this occurs for the cat's assumption in the range $\hat{c}_{\text{rat}} \in [0, 0.65]$. All this means is that the cat's policy for these assumption values are all identical.

The results in Figure 6.2 provide evidence in support of our hypothesis that when a player uses Bayesian estimates of their opponent's skill to inform their own policy, the expected reward for that player converges to the expected reward of a player that is given perfect knowledge of their opponent's skill.

The red line is the object of interest in this particular graph. We notice that the first point where the prior distribution is used to inform the cat's mixed optimal policy occurs at the point where the estimate of the rat's skill is $\hat{c}_{\text{rat}} = 0.5$ and the expected reward is approximately 0.182.

The red arrows on this line indicate the direction in which the player's expected reward has moved after each update of the posterior distribution. We see that this expected reward trajectory first gets worse for the cat moving up and to the right. This coincides with the posterior distribution after one and two games having more probability density for higher skilled rats. Since the estimator has more probability density over the higher, incorrect values for the rat's skill, we see the cat's expected reward move away from the optimal value of the Oracle.

After observing three more games however, the cat's estimate of the rat's skill starts to converge on the true skill of the rat. As the estimate converges, the expected reward of the cat also decreases toward the optimal expected reward. We see that after 5 games, the cat's estimate of the rat's skill has converged sufficiently to bring the cat's expected reward within the termination threshold of 0.001 of the estimator algorithm.

Next we observe the evolution of the estimator computed from the rat's perspective when estimat-

ing the cat's skill and assumption about the rat's skill.

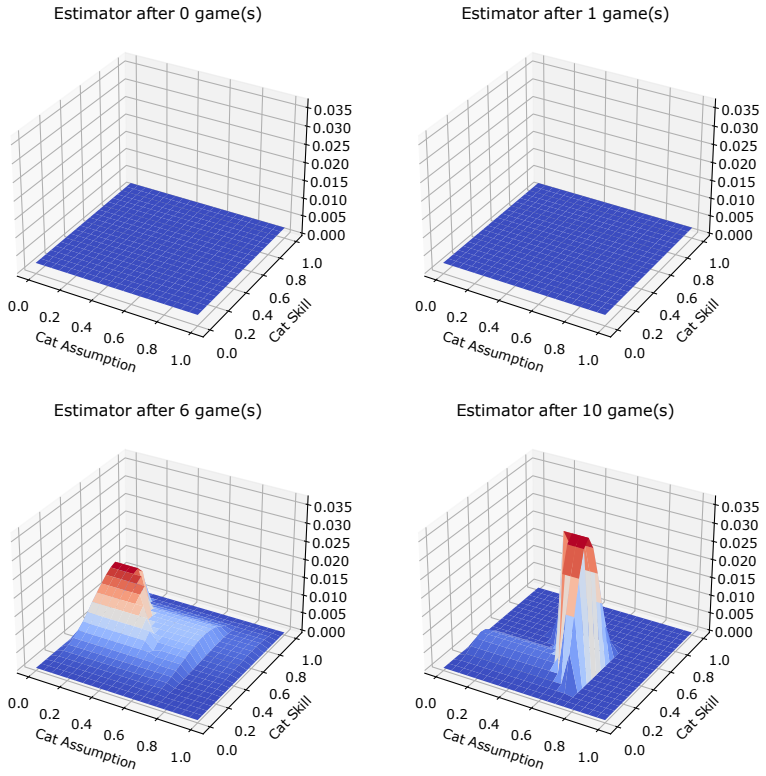


Figure 6.3: Posterior distribution of rat estimating cat in the contrived game.

In Figure 6.3 we notice that in the top two graphs the estimator's distribution has remained unchanged after 1 game of observed play. This suggests that in the first game the cat managed to perform actions that gave no information away to the rat.

After 6 games, in the bottom left graph, we observe that the rat's estimator has made some progress in estimating the cat's skill and assumption and is now more peaked along both axes. After 10 games, in the bottom right graph, we observe that the distribution has become far more peaked along both axes, with the peak coinciding with the true values of the cat's skill and assumption. The results in Figure 6.3 provide further evidence for our hypothesis that players can estimate skills from observed play.

Now that we have the rat's estimation distribution of the cat's skill and assumption we can compute the expected reward trajectory for a rat estimating the cat's skill.

In Figure 6.4 we observe that the static assumption curve is concave as expected for the rat. This curve is a slice of the expected reward function in Figure 5.7. The curve being concave means there is a maximum expected reward (or rewards) for which the rat, as the maximising player, is striving to achieve. We also notice that the Oracle point is at one point of maximum expected reward. We notice again that, in this discrete game, there are multiple values for the rat's assumption about the cat's skill, $\hat{c}_{\text{cat}} \in [0.3, 0.75]$, for which the optimal expected reward may be achieved.

Turning our attention to the expected reward trajectory we can see that the rat does manage to generally improve the performance of its mixed strategy as the skill estimate converges, but that it required more observed play to achieve this. We notice that the first simulated game provides no information to the rat about the cat's skill and assumption, and so the performance of the rat remains the same as if it had not observed the simulated game at all. The results in Figure 6.4 provide evidence

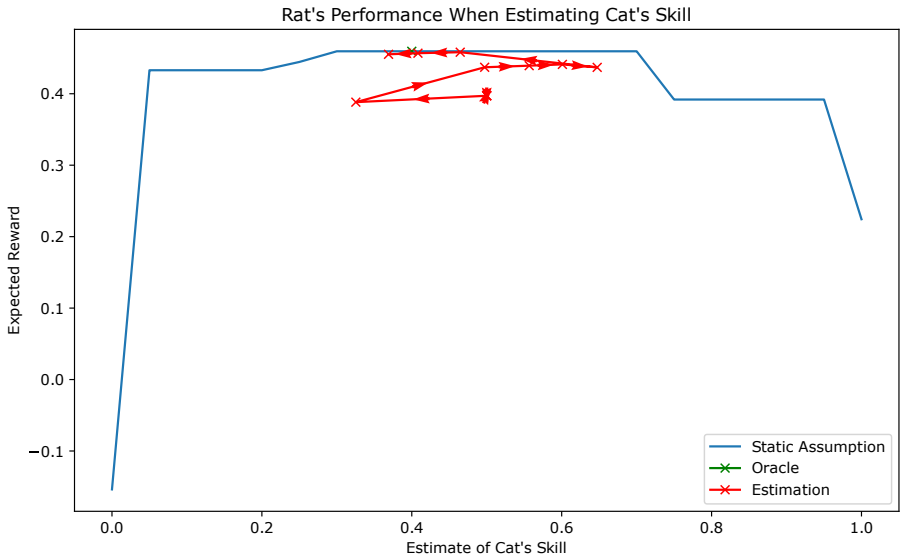


Figure 6.4: Expected reward of rat estimating cat in the contrived game.

in support of our hypothesis that skill estimation is valuable.

The results in this section demonstrate that our players are indeed able to estimate skill in our game environment. This estimation is done with varying levels of success based on the simulated play that is observed. In some simulated games, observed moves can provide useful information which allows the estimator to localise the true skill of their opponent very quickly. In other simulated games, the observed moves can provide little to no information about the opponent's skill, which leaves the estimator no better off than before observing these simulated games.

Furthermore, we notice that, in the case of the cat estimating the rat's skill, it is possible to observe play that can lead to a player's skill estimate localising on an incorrect value for the opponent's skill. Depending on the shape of the expected reward function, this can lead to worse performance from the player using skill estimation to inform their mixed strategy when compared with the uniform mixed strategy. The interested reader can find a more extreme example of this occurring in Figures A.3 and A.4 in the appendix.

Consider a game where both players are using skill estimation to optimise the performance of their mixed strategies. If players know when they are being observed by their opponent to update the opponent for the purpose of skill estimation, it could be beneficial for players to choose actions that give away as little information as possible to their opponent. Even better, if players could choose actions that skew their opponent's estimator in the wrong direction, such that the opponent's mixed strategy performs worse than if they had used the uniform mixed strategy. This would have interesting implications for players' ability to deceive their opponents or at least hinder their opponent's skill estimation efforts.

These observations open up a new level of reasoning for the players that is not explored in this research report, however, we believe it may be possible for adversarial players in this kind of setting to employ tactics that either slow down or mislead their opponent's estimation efforts in order to further maximise/minimise their own reward.

6.2 Estimating Skills with Increased Failure Factor

In this section we consider the effects of the failure factor, κ, on the skill estimation problem. Here we choose $\kappa = 0.01$ and perform the same experiments as in Section 6.1. Our expectation is that with the increased uncertainty in transition dynamics introduced by the positive failure factor, we observe that players take longer to localise the skill of their opponents.

Since the failure factor introduces positive transition probabilities for many observations that would previously have had zero probability (particularly for higher skilled players), we can understand why it would slow the rate of convergence of our estimator. There are now more explanations for observed moves than there were before.

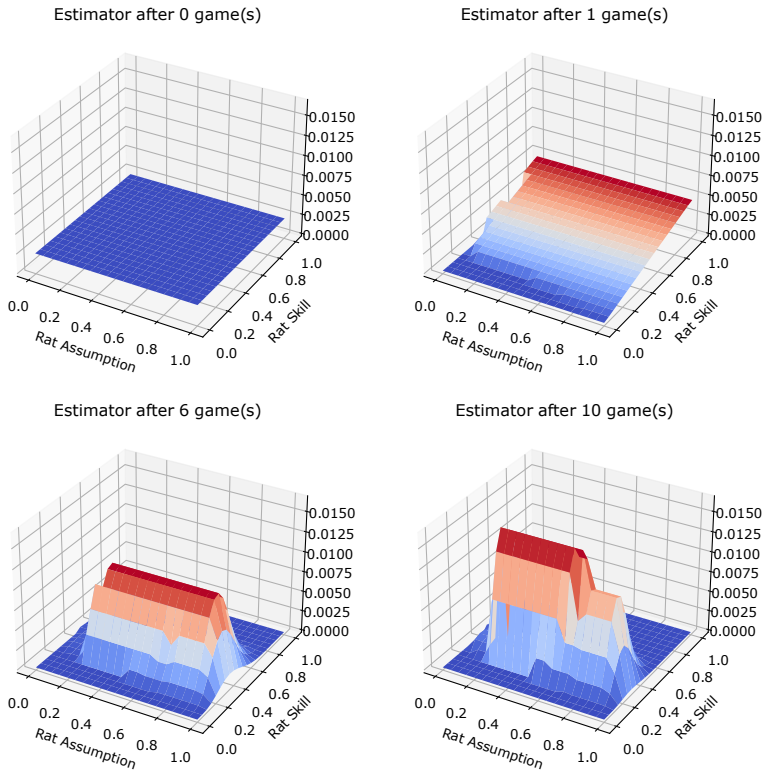


Figure 6.5: Posterior distribution of cat estimating rat in the contrived game with $\kappa = 0.01$.

In Figure 6.5 we can see that the cat takes much longer to localise the skill of the rat than it did previously in Figure 6.1. Previously, the cat was able to achieve a sufficiently good skill estimate of the rat within 5 simulated games to be able to perform within a reasonable threshold of the Oracle.

This time, even after 10 games the cat still does not have a good enough estimate to achieve a comparable expected reward to that of the Oracle. We do still observe the estimated distribution of the rat's skill and assumption converging, particularly along the skill axis more than along the assumption axis. The results in Figure 6.5 provide evidence in support of our hypothesis that players can estimate skills from observed play even when the failure factor, κ is positive.

We observe this phenomenon more clearly in Figure 6.6 where, again, the cat is initially mislead by the rat's play and its mixed strategy performs worse than the uniform mixed strategy. Like before in Figure 6.2 the cat's performance improves after observing the third and fourth simulated games before it worsens again. After observing 10 simulated games the cat's estimate of the rat's skill is starting to converge, but due to the altered shape of the expected reward function, we require the cat's estimate

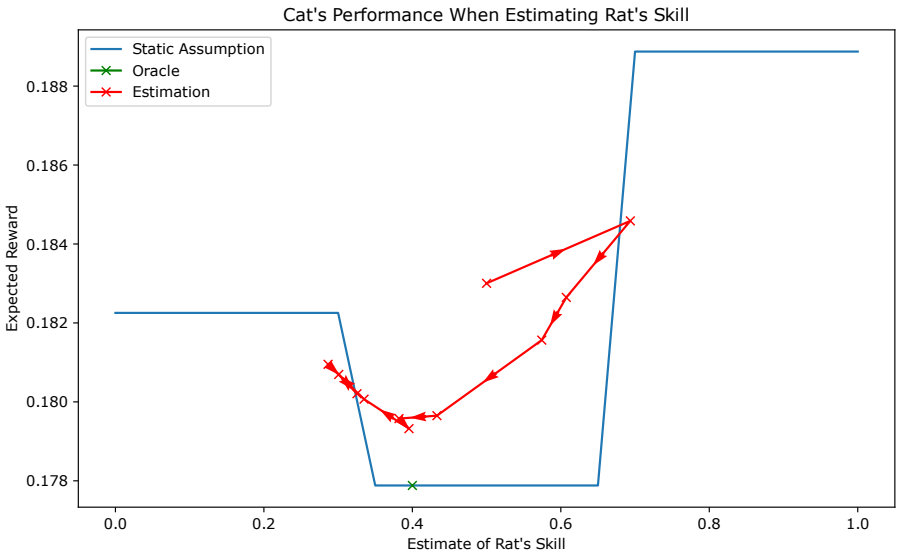


Figure 6.6: Expected reward of cat estimating rat in the contrived game with $\kappa = 0.01$.

to be even more accurate than before in order to achieve a comparable expected reward to the Oracle. The results in Figure 6.6 provide evidence in support of our hypothesis that skill estimation is valuable even when the failure factor, κ is positive.

Turning our attention to the rat we can see in Figure 6.7 that the rat's estimation of the cat's skill and assumption still converges rather slowly. Comparing the distributions to those in Figure 6.3 we see that again the first simulated game provided the rat with no information about the cat's skill or assumption.

We observe that after games 6 and 10 that the estimate is starting to converge, albeit to a lesser extent. Looking at the values on the z-axis and comparing between the two graphs we observe that the distribution after 10 games in Figure 6.3 is more peaked, with a maximum value of 0.035, when compared with the distribution in Figure 6.7 with a maximum value of 0.012. The results in Figure 6.5 still provide evidence in support of our hypothesis that players can estimate skills from observed play even when the failure factor, κ is positive.

When looking at Figure 6.8 we observe that the profile of the expected reward function is different to that in Figure 6.4 as expected due to the increased failure factor κ. We can now see the effect this has had on the rat's performance when using its estimate of the cat's skill to inform its mixed strategy. Before, the rat was able to perform fairly close to the level of the Oracle, however, in Figure 6.8 we see that the increased failure factor has affected the performance of the rat for the worse.

We observe in this section that for both the cat and the rat, the player's estimates of their opponent's skill take longer to converge to the point where player's can achieve expected rewards comparable to the Oracle. What we observed was that neither player was able to get sufficiently close to the Oracle's performance with only 10 simulated games worth of observed play. However, the results in Figure 6.8 still provide evidence in support of our hypothesis that skill estimation is valuable even when the failure factor, κ, is positive.

This under-performance compared to the previous section can be attributed to two factors. Firstly, the differing shapes of the expected reward function under an increased failure factor has resulted in players being penalised to a greater extent for inaccurate estimates of their opponent's skill.

For comparison, we observe in Figure 6.2 that if the cat's estimator has probability mass for skills

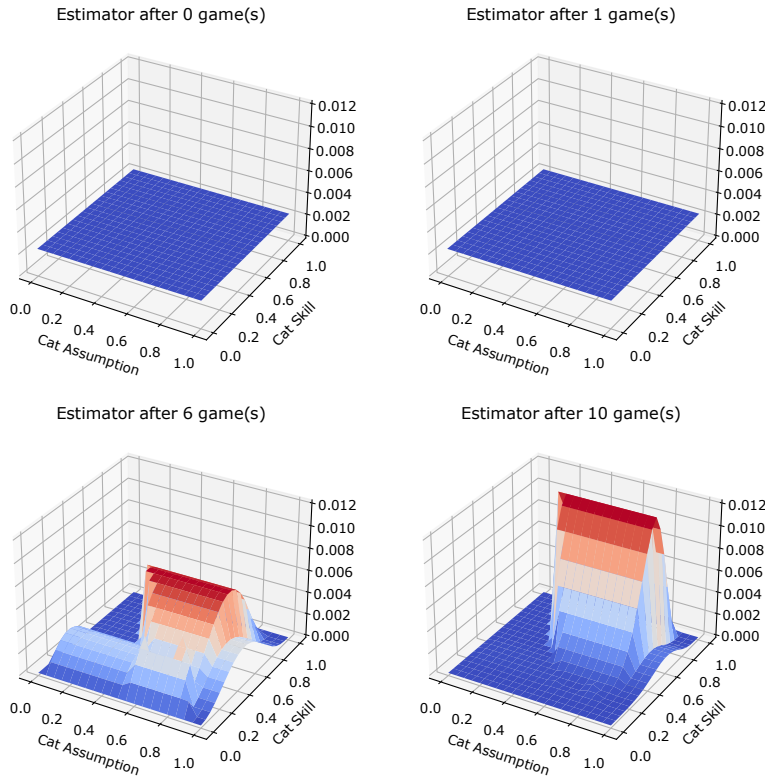


Figure 6.7: Posterior distribution of rat estimating cat in the contrived game with $\kappa = 0.01$.

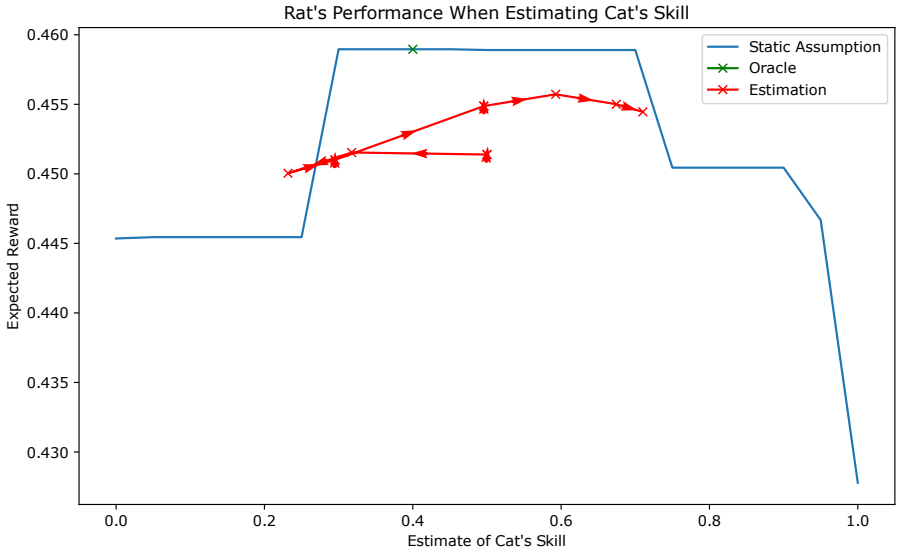


Figure 6.8: Expected reward of rat estimating cat in the contrived game with $\kappa = 0.01$.

in the range $[0, 0.65]$ and is zero elsewhere, the mixed strategy is able to perform as well as the Oracle. However, in Figure 6.6 this changes to the cat's estimator needing mass only in the range $[0.35, 0.65]$ and zero elsewhere in order to perform as well as the Oracle. This is purely due to the change in the shape of the expected reward function where $\kappa = 0.01$.

Secondly, the estimators themselves take longer to converge due to the increased number of explanations for observed play when computing the likelihood. This is again due to the positive failure factor, κ, and causes the probability mass to be spread more widely for the same observed play than

when $\kappa = 0$.

The consequences of a positive failure factor are that the players' estimates of their opponent's skills take longer to converge to the true value of those skills, and that this impacts the performance of the players when using a mixed strategy based on this estimate.

We notice that while the player performance is impacted negatively by choosing a positive failure factor, the players are still able to perform better on average than if they had stuck with the uniform mixed strategy. This suggests that even in this setting it is still advantageous for players to estimate their opponent's skills.

6.2.1 The Convergence of Skill Estimates and Reaching Equilibrium

In this subsection we discuss the ability of a player's estimate to localise the true value of the opponent's skill. We also discuss whether the game reaches a state wherein both players attain their optimal rewards, despite having started from a circumstance in which one player has a large degree of uncertainty. Such cases show that information has been extracted through play, that this knowledge improves an agent's play, and it is sufficient to cause an agent to evolve from sub-optimal to optimal play.

In the next experiments the cat is given knowledge of the rat's skill at the beginning of the game (i.e. the cat is an oracle player). We do this to clarify the dynamics of estimation by freezing one player in the oracle state and allowing the other player to estimate and adjust its strategy. The cat will perform the optimal actions for its skill level with perfect knowledge of what the rat's optimal strategy should be. The rat will perform sub-optimally at first as it has no knowledge of the cat's skill and will have to estimate it during play. As the rat's estimate of the cat's skill begins to localise on the true skill value we expect to observe the expected reward of the rat increase.

In the first experiment for this section, we consider an example where the expected reward function of the rat is maximised only when it correctly identifies the cat's skill. In this experiment, the cat's skill $c_{\text{cat}} = 0.1$, the rat's skill $c_{\text{rat}} = 0.3$, and the cat's assumption about the rat's skill $\hat{c}_{\text{rat}} = 0.3$ (i.e. the cat has perfect knowledge of the rat's skill).

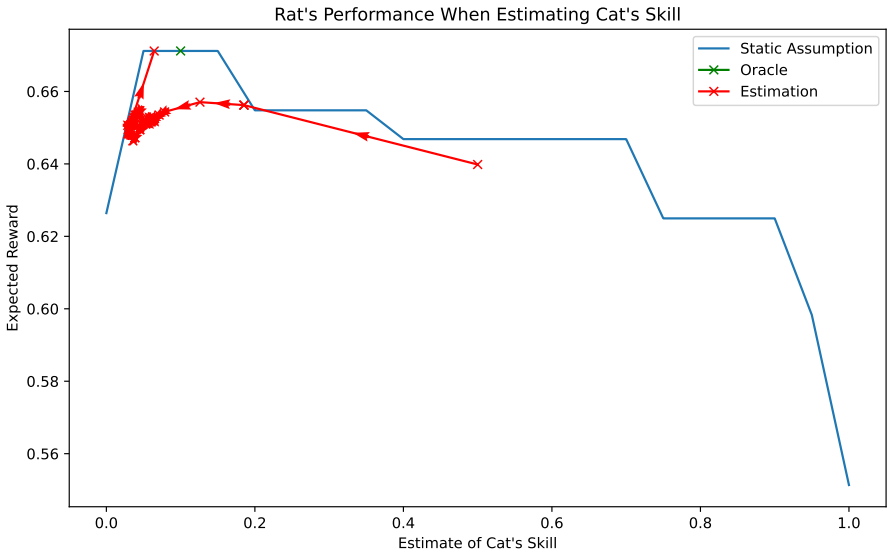


Figure 6.9: Maximising the rat's performance when the cat's skill is estimated correctly.

In Figure 6.9 we see that the expected reward function is maximised when the rat's estimate of the cat's skill localises on the true value of the cat's skill (which is the value 0.1). We also notice that estimation path eventually converges to the maximum value of the reward function once the rat's estimate of the cat's skill has localised sufficiently to exclude the possibility of the cat's skill being either 0.0 or 0.2.

It is also worth noting that the rat's estimate of the cat's skill spent a significant number of iterations around a sub-optimal point where the estimator still had non-zero probability for other skill values of the cat.

This is illustrated in Figure 6.10 where we see that after 1 game, the estimate has been able to localise slightly, with the most probability mass in the red area of the graph. After 24 games, the probability mass has concentrated further around the cat's true skill value of 0.1. Finally, after 46 games, the probability mass spikes around the cat's skill value of 0.1 and there is zero probability mass assigned to skill values of 0.0 or 0.2, which allows the rat to maximize its expected reward.

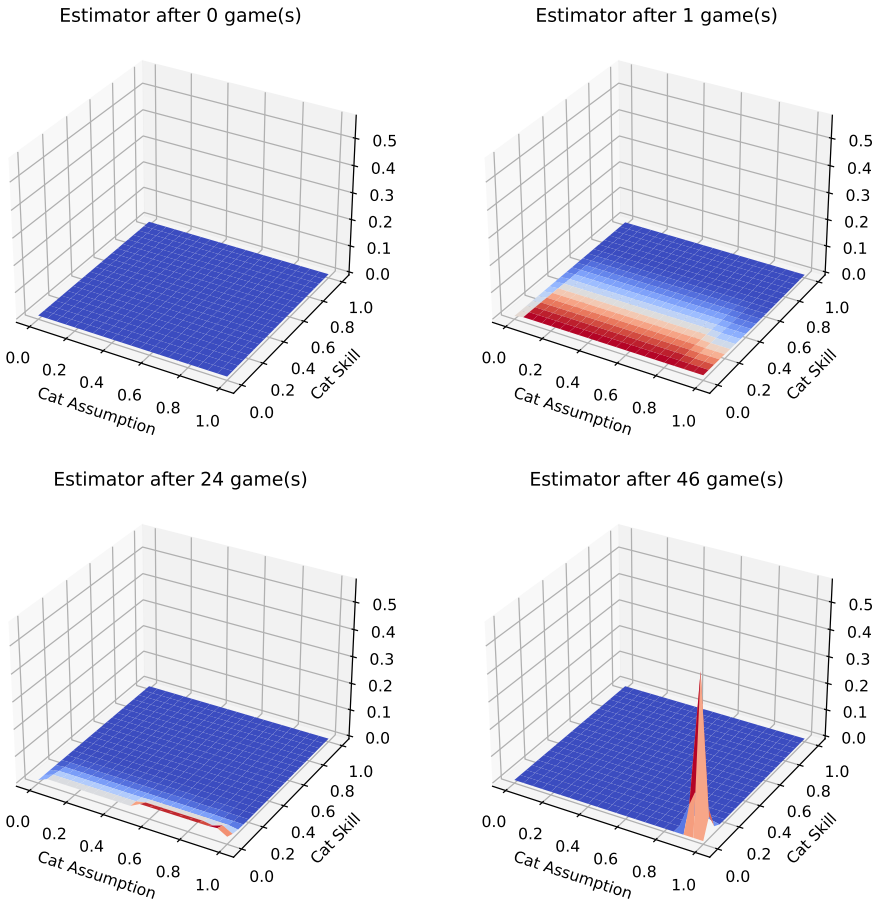


Figure 6.10: Localising the cat's true skill level.

We note that after 46 games, both the cat and the rat have sufficient knowledge of their opponent's skill to minimise/maximise their respective expected rewards. Each player is playing their optimal complete-knowledge policy, corresponding to their respective oracle strategies. Assuming both players continue playing their optimal strategies, further estimation by either player would be pointless as a more accurate skill estimate of their opponent would not change their strategy or expected reward at this point.

It is worth noting that in the above game, it appears the the strategies and expected rewards of the players have reached a type of equilibrium. As both players continue to employ their optimal strategies to attain their optimal expected rewards, this equilibrium will be maintained.

We now consider a game where, under certain circumstances, an opponent's play is indistinguishable for a range of skills. In this situation, the player estimating the opponent's skill would only be able to localise the opponent's skill level up to a point. In these circumstances, we expect to see the estimator converge sufficiently around the skill values for which the opponent's play is indistinguishable, but we do not expect to see a spike in the estimator around a particular skill level as above.

We also anticipate that the player's expected reward is maximised when the player's estimate of the opponent's skill has converged sufficiently around the skill values for which the opponent's play is indistinguishable. Since the opponent's strategy is indistinguishable in this range of skill levels it makes sense that the player's best response strategy would also be indistinguishable for beliefs about the opponent's skill in this range.

In this final experiment for this section we consider a specific setup of the contrived game where the maximal expected reward for the rat is achievable for multiple values of the rat's estimate of the cat's skill. In this experiment, the cat's skill $c_{\text{cat}} = 0.6$, the rat's skill $c_{\text{rat}} = 0.5$, and the cat's assumption about the rat's skill $\hat{c}_{\text{cat}} = 0.5$ (i.e. the cat has perfect knowledge of the rat's skill).

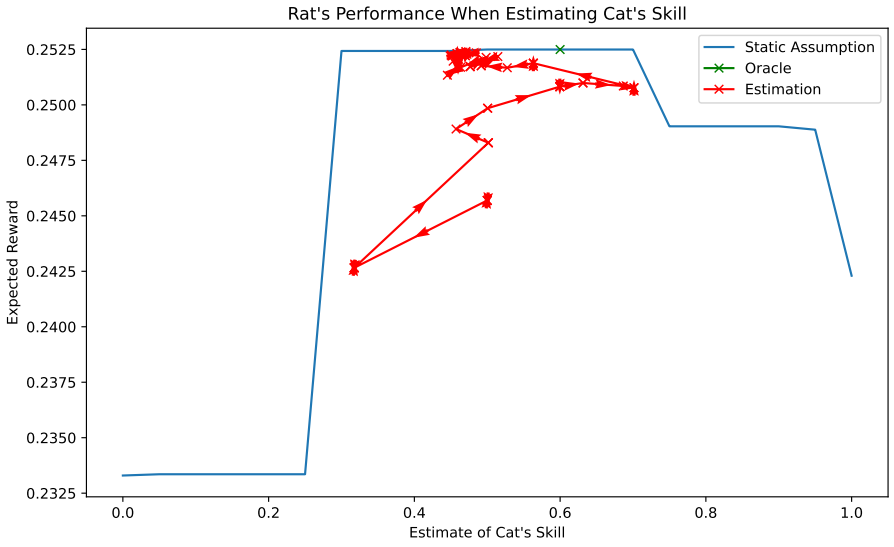


Figure 6.11: Maximising the rat's performance for a range of estimates for the cat's skill.

In Figure 6.11 we see that the rat's expected reward is maximised when $\hat{c}_{\text{rat}} \in \{0.3, 0.4, 0.5, 0.6, 0.7\}$. Consider that, for this set of beliefs of the rat about the cat's skill, the expected reward and optimal policy is unique, and insensitive to the rat's belief within this set. As such, we claim that this set of beliefs could be collapsed into a single combined belief that represents the set without affecting the outcome of the game or the optimal policy of the rat.

In Figure 6.12 we observe that after 49 games the rat's estimate of the cat's skill has localised to the range described above. Since the rat's optimal policy at this point is unique for this range of beliefs about the cat's skill level, the estimate does not need to localise further for the rat to achieve the maximal expected reward against the cat.

It is important to note that these empirical results demonstrate the feasibility of using the outcomes of play to reduce (and ultimately identify) variables describing an opponent; they are not claimed to

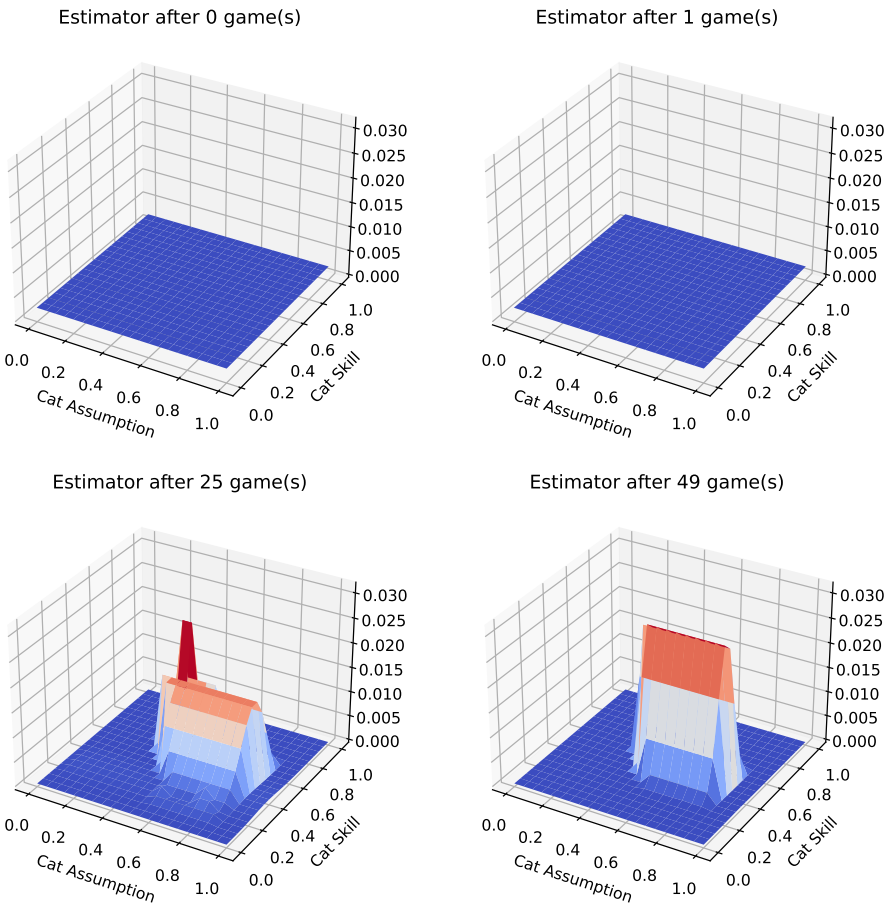


Figure 6.12: Localising the cat's skill level when more than one value achieves the maximal expected reward for the rat.

be Sequential Equilibria [25]. Firstly, one of the agents begins and maintains full knowledge, so the symmetry is broken in the scenario we present; secondly, what is illustrated in Figure 6.11 and Figure 6.12 is the transition toward convergence of beliefs. Recall, the emphasis of this research report has been on an external observer, these instances of estimation during play are provided to show this is indeed feasible too.

6.3 The Validity of Assuming Rationality When Estimating Skill

Recall that Selten's [23] Perfect Equilibrium extends the traditional notion of Nash Equilibrium by incorporating the idea of sub-game perfection, which requires that not only must each strategy be a best response to the other players' strategies, but this condition should hold true at every possible decision point within the game. A strategy profile is a Selten Perfect Equilibrium if it constitutes a Nash Equilibrium in every possible sub-game of the larger game. However, in order to address the issue of some sub-games being off the equilibrium game path, and thus having no support (i.e. a zero probability of reaching that sub-game), Selten introduces the concept of the "trembling hand" so that all sub-games have positive support.

Kreps and Wilson's [25] Sequential Equilibrium is a refinement of Selten's Perfect Equilibrium that addresses certain deficiencies in the latter. While Selten's Perfect Equilibrium requires strategies to be sequentially rational, it does not explicitly account for beliefs about an opponent's strategies off

the equilibrium path. In contrast, Kreps and Wilson's Sequential Equilibrium incorporates the notion of consistency in players' beliefs throughout the game. This refinement helps to address issues related to off-equilibrium beliefs and provides a more robust framework for analyzing sequential games with imperfect information.

Our work is closer to that of Kreps and Wilson's Sequential Equilibrium in that we also have a concept of players' beliefs in our framework. Kreps and Wilson state, "We have two motives for proposing this alteration of Selten's definition. The first is pragmatic: In many examples of interest, it is vastly easier to verify that a given equilibrium is sequential than that it is perfect. Second, making explicit the construction of beliefs off the equilibrium path enables discussion of which beliefs are 'plausible' and which are not."[25]

Like Kreps and Wilson, the introduction of skill and beliefs into our framework is also motivated by the desire to make the construction of beliefs explicit. However, and perhaps more importantly, we introduce the concept of beliefs into our extended form game to enable us to explore the question of skill estimation. Unlike Kreps and Wilson, we are not concerned with verifying whether the solutions to our game are sequential or perfect equilibria.

In our framework thus far we have been considering players who play optimally. It makes sense that players can assume their opponent is going to be playing rationally since, if this assumption is violated, the opponent is irrational, their expected reward is bounded above by that of a rational player. Put simply, if a player plays irrationally, it can only harm their expected reward. This assumption of rationality is common in the field of game theory and is the basis of many of the results in this field [2] [22] [25].

As such, when developing our algorithm for the skill estimator, we make the assumption that the opponent is acting rationally. This is may be a reasonable assumption for a player to make if the opponent is not concerned about the player's skill estimation and seeks only to optimise its immediate expected reward. However, consider the third party observer that is only interested in estimating a player's skill. In this situation, there may not be a strong justification to assume that players act rationally for the purposes of estimating their skill and when expected reward is not a consideration.

In the next section we conduct experiments that assess the validity of the assumption made by the estimator that players are rational. We do this by introducing the concept of an irrational player that selects actions uniformly at random. In this section we assess the robustness of our skill estimation algorithm against this irrational player. This allows us to assess the validity of the rationality assumption that has been incorporated into the skill estimation algorithm.

In these experiments we expect to observe some simulated games where the irrational opponents inadvertently give away more information about their skill than they would if they were playing rationally. We also expect to observe some simulated games where the irrational opponents give away less valuable information about their skill than if they were playing optimally.

In Figure 6.13 we observe the cat's estimates of the rat's skill and assumption based on simulated games where the rat follows a uniform random policy. It is interesting to note that in this instance, the rat's random play has given away more valuable information to the cat than its optimal play did in Figure 6.1 after the first simulated game. However, if we compare the change in the skill estimates from 1 game to 3 games, we observe that the random rat has given away less valuable information about its skill in games 2 and 3 than the optimal rat did. In Figure 6.13 the cat's estimate shows a rapid shift after the first game while not moving significantly in the second or third games.

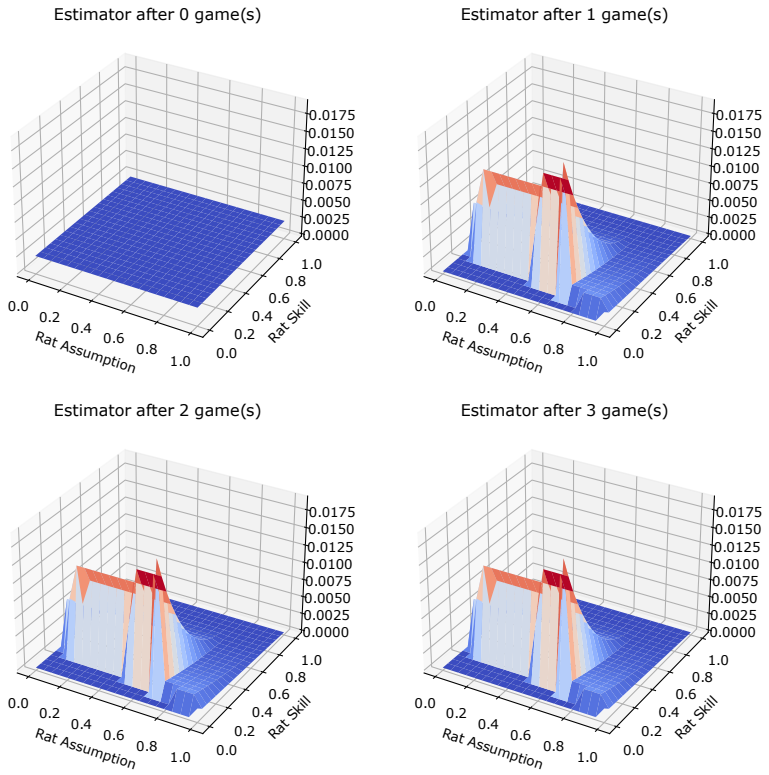


Figure 6.13: Posterior distribution of cat estimating an irrational rat in the contrived game.

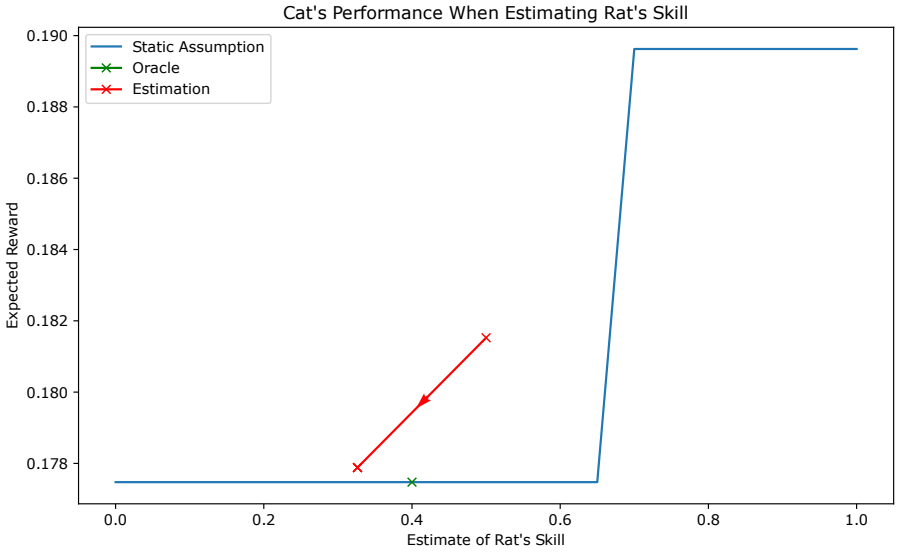


Figure 6.14: Expected reward of cat estimating random rat in the contrived game.

This suggests that the cat observed a lucky first game where the random rat inadvertently gave away some valuable information about its skill. In the second and third games the cat was not so lucky, and the rat did not give away any valuable information about its skill. This confirms what we expected about there being some games where a random player inadvertently gives away more valuable information than an optimal player would, and there are some games where the random player gives away less valuable information than an optimal player.

When looking at the cat's performance in Figure 6.14, we observe that the cat's expected reward

takes a significant step towards the point of optimal expected reward after the first simulated game. This is driven by the fact that the cat observed a lucky first game which gave it sufficient information about the rat's skill to allow the cat's mixed strategy to achieve an expected reward close to that of the Oracle.

The fact that simulated games 2 and 3 provided the cat with no further information about the rat's skill has not affected the performance of the cat's mixed strategy. Had it not been for that first lucky game, it is unlikely that the cat would have performed as well after so few observed games. If we compare this graph to that in Figure 6.2, we can see how the optimal rat did not provide the cat with sufficient information for its performance to improve to near optimal until after game 5.

The above experiments illustrate the outcome of the cat estimating the rat's skill over only one trial of the games played. Due to the random elements of the game, we conduct an additional experiment where the cat estimates the rat's skill over multiple trials of independent learning. In this experiment the players play 3 rounds of the game, with the cat estimating the rat's skill from the observed play in those 3 rounds, which we will call the first trial. We then reset the cat's estimate to the uniform prior and have the players play another 3 rounds, which will be the second trial. We repeat this for a total of 20 trials, and then average the cat's estimates of the rat's skill across the 20 trials.

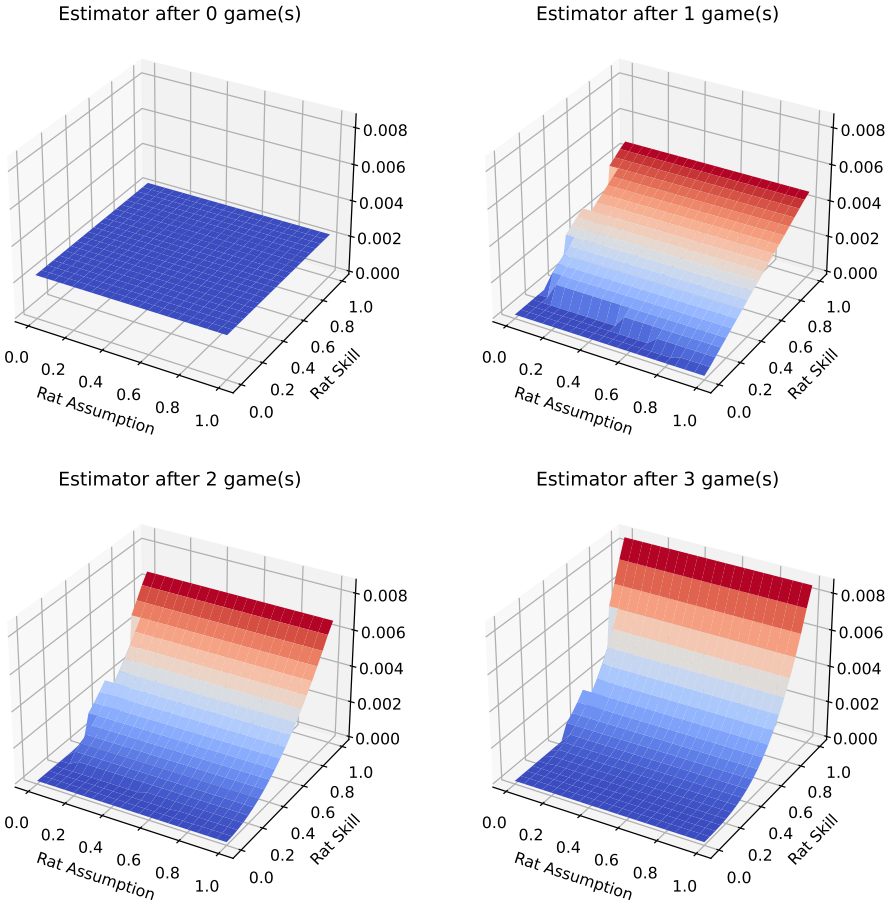


Figure 6.15: The average of the cat's estimate of the rat's skill after 20 trials.

The average evolution of the cat's estimate of the rat's skill in Figure 6.15 does adjust slightly across games 1, 2, and 3. This suggests that when playing against an opponent that selects moves randomly, there is some information, on average, being given away by the rat that selects actions at random. Indeed, if the cat is playing its optimal strategy, and the rat is selecting moves at random,

we expect the cat to win more quickly than if it were playing against a rat that plays rationally.

When observing the underlying play, the cat tends to win the game after only a few moves, and in that time, the rat has a limited number of randomly selected moves to make, many of which do not provide much information about its skill level that would affect the cat's estimate of the rat's skill. This highlights the point above that when the cat is lucky enough to observe informative play from the rat, and improve its estimate of the rat's skill, the performance of the cat's mixed strategy will improve.

In the contrived game, the moves immediately available to the rat are either low skilled moves that give no information away, or very high skilled moves that suggest to the skill estimation algorithm that the rat is following a high skill rational strategy. On the rare occasion that the rat is able to successfully make a high skilled move, the cat's estimate is biased towards the higher end of the skill axis.

It is interesting to observe that certain play can bias the estimation algorithm away from the true value of the skill being estimated. This bias in the cat's skill estimate does not affect the win rate of the cat as the cat is actually winning more frequently than it would against a rational rat. Playing irrationally does not benefit the rat in terms of expected reward, but it does have a biasing effect on the estimation algorithm.

We now observe the case where the rat estimates the skill of a cat that follows a uniform random policy. In Figure 6.16 we see that the rat has been unlucky and that even after 10 games, the random cat has not given away any valuable information about its skill. The rat's estimate remains the same as the uniform prior it started with even after observing 10 simulated games.

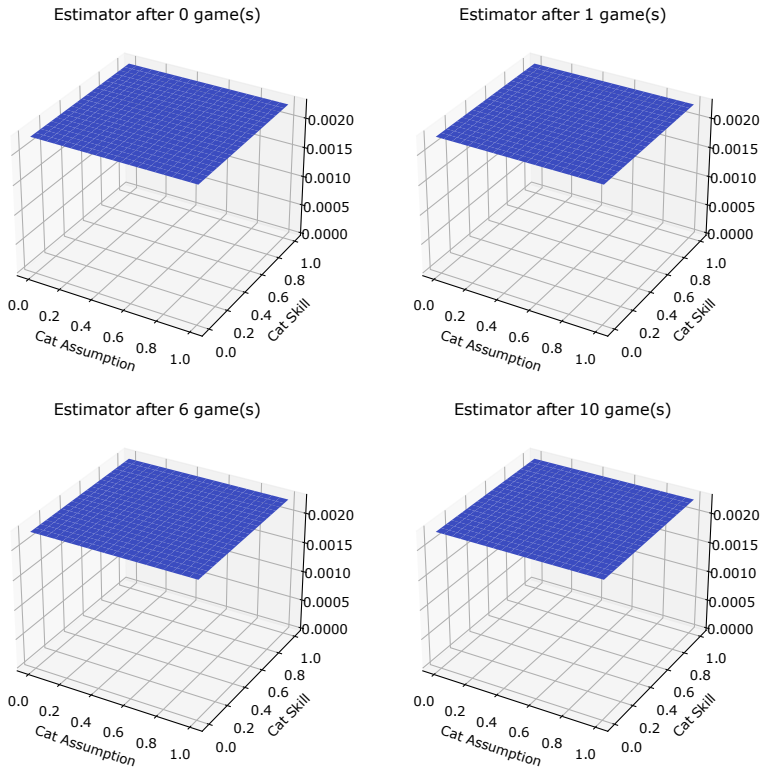


Figure 6.16: Posterior distribution of rat estimating random cat in the contrived game.

This happened because the random cat always chose actions that moved it along edges with weight

0. It makes sense that the rat would not be able to make any inferences about the cat's skill if these are the only moves that the rat observed the cat making. We do note that the rat won all of these simulated games.

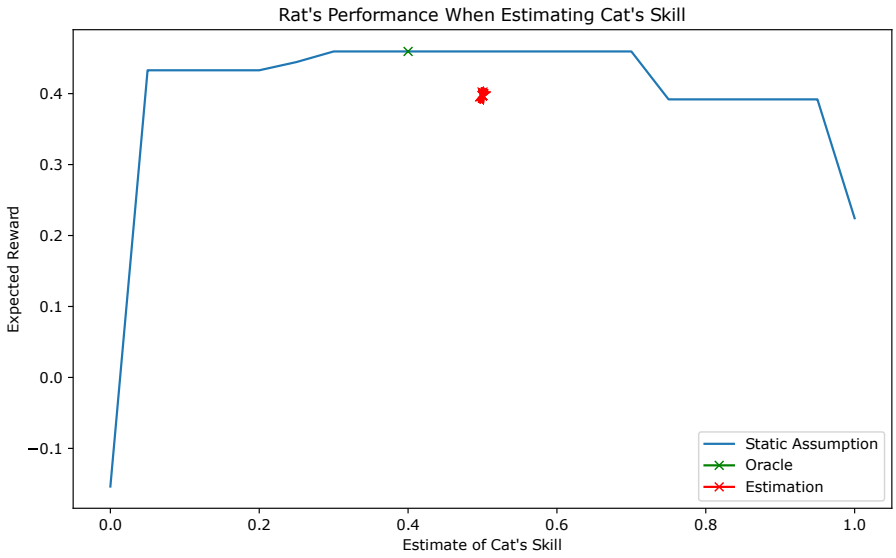


Figure 6.17: Expected reward of rat estimating random cat in the contrived game.

Unlike the previous experiment where the cat got lucky with the moves it observed, in this experiment the rat got unlucky with the moves it observed. Since the rat's estimator has not changed from the uniform prior it started with, we expect to observe in Figure 6.17 that the rat would not perform any better than had it used the uniform mixed strategy. When we compare this to Figure 6.4 we notice that even though the rat's estimate of the optimal cat's skill converged slowly, the rat was still able to perform better than the uniform mixed policy after 10 games of observed play.

We again consider the evolution of the rat's estimate of the cat's skill when we average over multiple trials. Again we allow the rat to perform estimation on the random cat over 20 trials with 3 games per trial.

In Figure 6.18 we observe that the rat is able to glean some information about the cat from these games, but not enough to be able to localise the cat's skill very accurately. Again there is some change in the shape of the average estimator surfaces after each successive game. However, unlike in Figure 6.16, we do see that cat is revealing some information about its skill in some of the games. This confirms the observation above that the rat was unlucky in the games played against the cat in that it did not observe the cat making any informative moves. As with Figure 6.15, we see in Figure 6.18 that the rat's estimate of the cat's skill is also being biased by the irrational behaviour of the cat towards the higher end of the skill axis and for the same reason.

In this section we have observed that random play can lead to situations where the irrational player inadvertently gives away valuable information about its skill. It can also lead to situations where the irrational player gives away less information than the optimal player does. When a player behaves irrationally, it becomes more difficult for the skill estimation algorithm to localise on the player's skill. When the rationality assumption of the estimation algorithm is violated, the skill estimate can be biased by unlucky/lucky runs, especially when there are fewer games and/or trials over which to observe the irrational players actions.

In this chapter we observed in Section 6.1 that players can estimate opponent skills based on

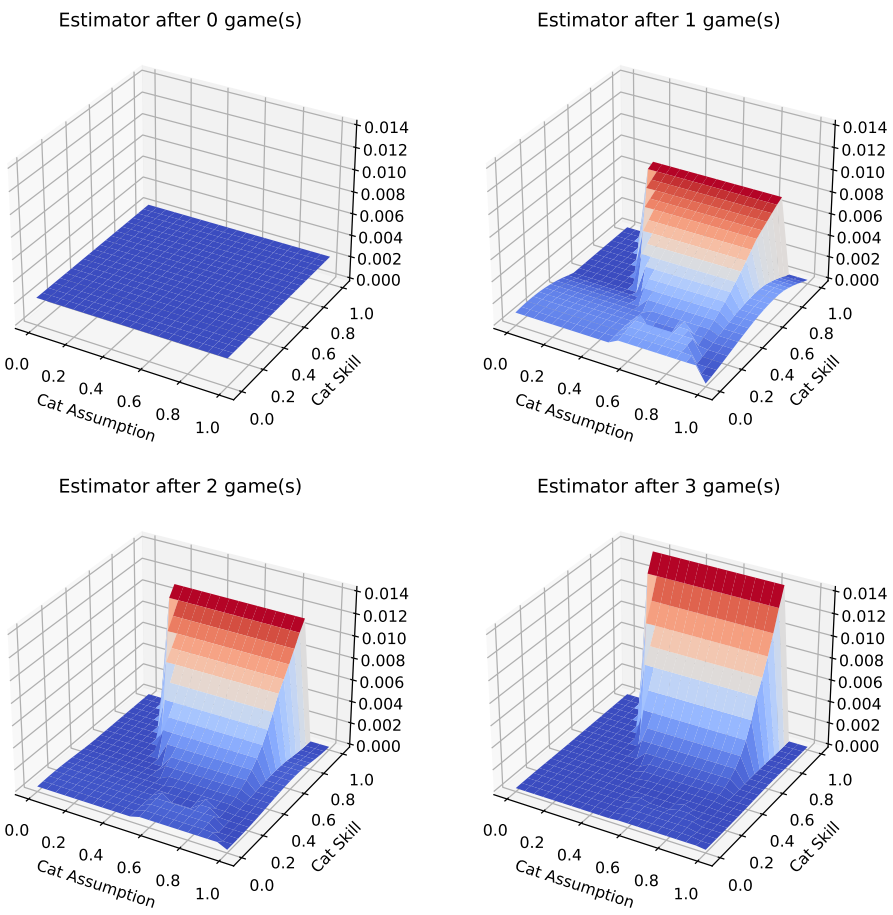


Figure 6.18: The average of the rat's estimate of the cat's skill after 20 trials.

observed play and that using this estimate to inform a mixed optimal policy, were able to achieve expected rewards approaching that of the Oracle player. This provides evidence in support of two of our hypotheses that players can estimate skill from observed play of rational opponents, and that skill estimation is valuable.

In Section 6.2 we observed that choosing a positive value of the failure factor, κ, has a negative effect on the performance of skill estimation. We observed that estimation took longer to converge and that the expected rewards achieved by the mixed optimal policy took longer to approach that of the Oracle player. Players were still able to perform better by estimating than using a uniform mixed policy. This provides further evidence in support of two of our hypotheses that players can estimate skill from observed play and that skill estimation is valuable even when the failure factor, κ, is positive.

In Section 6.3 we tested the robustness of our estimation algorithm to irrational players. We observed that players that use a uniform random policy have mixed effects on the estimation algorithm. In some instances it made the estimation task easier, and in other instances it made estimation more difficult. This observation raises an important question about the ability of players to influence the skill estimation algorithm by carefully selecting sub-optimal moves. This is potentially an area of consideration for future works where a player may be interested not only in selecting actions that maximise its expected reward, but is also interested in selecting actions that minimise the information gained by an estimator trying to estimate its skill.

Chapter 7

Conclusion

In this research report we explore the problem of skill estimation in discrete pursuit-evasion games. In the literature there are examples of skill estimation and opponent modelling however the area of integrating skill estimation into opponent modelling is still "under-explored" [9]. We use the world of adversarial, two-player, zero-sum, time-restricted, pursuit-evasion games as a basis for exploring opponent modelling and skill estimation.

By doing this we were able to test the following hypotheses presented in Chapter 4:

1. Incorrect assumptions cause harm: The expected reward obtained by a player making an incorrect assumption about their opponent's skill should always be less than or equal to the expected reward when making the correct assumption about the opponent's skill.

2. Players can estimate skills from observed play: When players use Bayesian inference in estimating their opponent's skill their posterior distribution concentrates probability mass around the true value of the opponent's skill when sufficient play is observed.

3. Skill estimation is valuable: When a player uses Bayesian estimates of their opponent's skill to inform their own policy, the expected reward for that player converges to the expected reward of a player that is given perfect knowledge of their opponent's skill.

In Chapter 4 we devised a research framework in which we could perform experiments that would allow us to test these hypotheses. As part of this framework we devised a Markov game environment which represents a subset of discrete, zero-sum, pursuit-evasion games. In this game environment we were able to use the Expectiminimax algorithm to compute an expected reward function in closed form. We also provided a mechanism by which players can use Bayes Theorem in this game environment to estimate the skills of their opponents.

In the sections that follow we detail the conclusions reached from the experiments conducted in Chapters 5 and 6. We also point out where the experiments have provided evidence in support of or against our hypotheses.

7.1 The Expected Reward Function

In Chapter 5 we have observed the results of a series of experiments where we compute the expected reward function for several instances of our Markov game and make some observations based on certain

slices of this high dimensional surface. First we confirmed that the empirical expected reward does converge to the closed-form empirical reward for a sufficiently large number of simulated games.

Using the closed-form expected reward function we observed that the function is monotonically increasing along the rat's skill axis and monotonically decreasing along the cat's skill axis. We also observed that the function is concave along the rat's assumption axis and convex along the cat's assumption axis. Players attain the optimal expected reward for a given set of skills when their assumptions match the true values of their opponent's skill. This last point provides evidence in support of Hypothesis 1 that incorrect assumptions cause harm. This feature of the expected reward function was observed in all experiments in our game environment.

In the experiments involving a positive failure factor, κ, we observed that in general the expected reward for both players decreased. These experiments also showed that players may be encouraged to play more conservatively given the increased uncertainty in the transition dynamics. This conservative play resulted in players being less severely punished for having assumptions about their opponent's skill that were extremely wrong.

In the experiments where players discounted the reward (i.e. $\gamma < 1$) we observed that players became more risk-seeking, in some cases preferring shorter, more uncertain paths to a reward over longer, more certain paths to a reward. This risk-seeking behaviour can undermine the effects of the player's assumption about their opponent's skill on the expected reward function.

Finally, in the experiments where the scoring was altered and we chose $\beta = 0$ we observed that in some games this made the players more conservative. It also had the effect of changing the shape of the expected reward function significantly across certain axes.

7.2 Skill Estimation

In Chapter 6 we have observed the results of a series of experiments where players estimate their opponent's skill based on observed play and use this estimate to inform their mixed optimal policy. The experiments on skill estimation did provide evidence in support of Hypothesis 2 and demonstrated that players can indeed estimate skill successfully in this game environment.

The resulting performance of the mixed optimal policy when estimating the opponent's skill performs better than a uniform mixed policy provided the skill estimate has converged sufficiently. As the skill estimate converges to the true skill of the opponent, the expected reward from the mixed optimal policy converges to the expected reward of the Oracle player who is given perfect knowledge of the opponent's skill. The results of these experiments have provided evidence in support of Hypothesis 3 and confirm that skill estimation is valuable.

The effects of a positive failure factor, κ, on the performance of estimating players is twofold. Firstly, the increased failure factor leads to slower convergence of skill estimates given that the probability mass associated with observed moves is distributed more widely, particularly for higher skilled players. Secondly, the changes observed in the expected reward function for positive κ have the effect of reducing the effectiveness of the mixed optimal policy when the skill estimates have not converged sufficiently. On average, players that estimate skills still perform better on average than players who use a uniform mixed policy when $\kappa > 0$.

Finally, the effects of random play by the opponent on the skill estimates of players can be unpredictable. In some situations, random players may inadvertently give away more valuable information

about their skill than an optimal player would. In other situations, random players may give away less valuable information about their skill than an optimal player would.

We noticed through these experiments that some action choices by an opponent provide more information to an estimating player than other action choices. If players were to play opponents that were also using Bayesian inference to estimate the player's skill, it may be possible to select a mixed policy that not only approaches optimal performance against the given opponent, but also misleads the opponent's estimation of the player's skill to induce sub-optimal performance from the opponent. Such policy construction is not explored in this research report but is recommended as an important piece of future work to be conducted.

7.3 Future Work

When conducting our experiments in the preceding chapters, there were some results that raised some new questions. The following points detail possible areas of future work that can build on the work presented in this research report:

- We observed that in this game environment players should always choose improving their skill over improving the accuracy of their assumption about their opponent's skill. It would be interesting to know if there are games for which this choice would be reversed.

- We introduced a failure factor parameter, κ, into our game environment in order to introduce transition uncertainty for players of all skills. This helped us test the skill estimator more robustly since a positive failure factor ensured that there was positive support for failed actions for players of all skills. It would be interesting to see alternative ways of introducing a positive support for these actions. It may also be possible to introduce a sub-optimality parameter that makes players choose other sub-optimal actions other than staying in the current state.

- When considering the effects of discounting on the expected reward function we noticed that players tended to become more risk-seeking in their optimal policies. It would be worth exploring in what settings this kind of behaviour may be desirable. It would also be interesting to explore how the value of the discounting factor γ affects the degree of risk-seeking and at what point, if any, the player could be considered to be acting sub-optimally by a player that does not perform discounting.

- When estimating opponents' skills and computing the performance of a mixed optimal policy informed by these skill estimates we noticed realisations of simulated games that gave no useful information, or in some cases misleading information. One particularly exciting piece of future work would be to devise a way for players to construct policies in which they account for their opponent's estimate of their own skill. In such a setting it is conceivable that players would prefer to take actions that limit the knowledge gathering of their opponent, or even better, mislead them such that they behave sub-optimally. From our experiments we see that this could be possible and would provide another dimension along which players can optimise their expected rewards.

Appendix A

Skill Estimation Graphs

A.1 Skill Estimation of Expectiminimax Players Continued

In this section we provide the relevant graphs of the players' skill estimation and performance on the random small game and random large game. Similar graphs are presented for the contrived game in Section 6.1.

In Figure A.1 we notice that after observing 10 games the cat has been able to localise the skill of the rat accurately but not the assumption of the rat about the cat's skill.

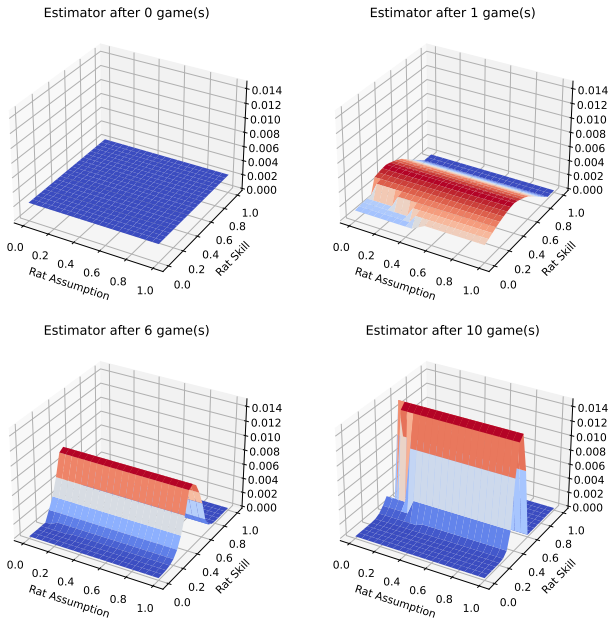


Figure A.1: Posterior distribution of cat estimating rat in the random small game.

In Figure A.2 we observe that the performance of the cat does improve as the skill estimate localises on the true value of the rat's skill and approaches the performance of the Oracle player.

In Figure A.3 we notice that after observing 10 games the rat has not been able to localise the skill of the cat or the cat's assumption about the rat's skill very accurately. Upon further investigation is

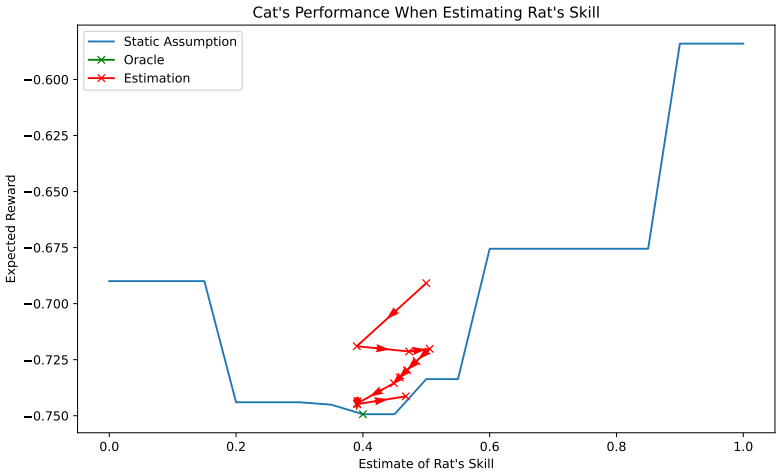


Figure A.2: Expected reward of cat estimating rat in the random small game.

was determined that the cat had particularly good luck in the first 4 games, which threw off the rat's estimation of the cat's skill. Thereafter, the transitions observed from the cat's play did not provide any further useful information to the rat.

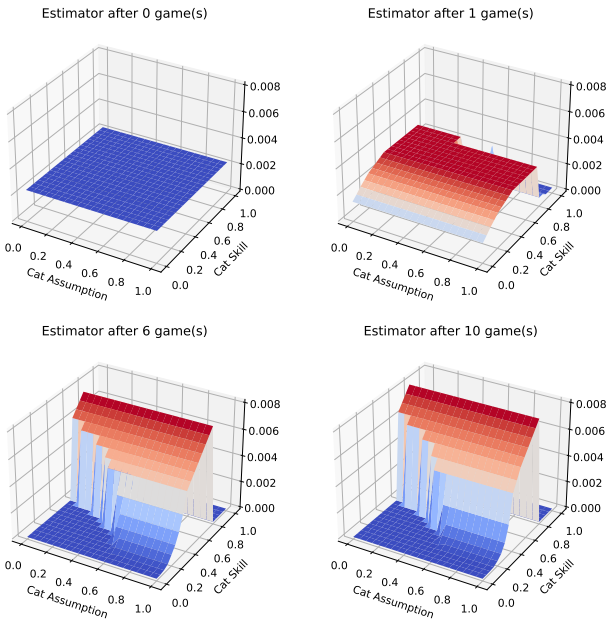


Figure A.3: Posterior distribution of rat estimating cat in the random small game.

In Figure A.4 we observe that the performance of the rat is worse after 10 games than a player using the uniform mixed policy. This is attributed to the cat having particularly good luck in the first 4 games and thereafter the rat gained no further useful information about the cat's skill.

In Figure A.5 we notice that after observing 10 games the cat has been able to localise the skill of

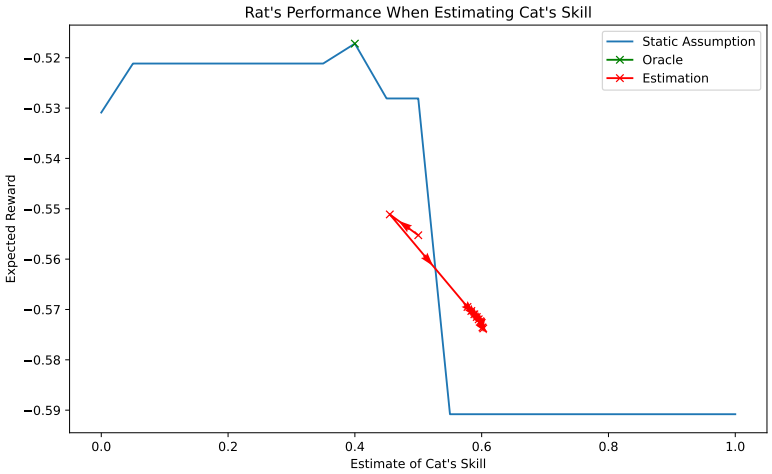


Figure A.4: Expected reward of rat estimating cat in the random small game.

the rat and the rat's assumption about the cat's skill very accurately. We can clearly see the posterior distribution's peak occurs at 0.4 along the rat skill axis, and 0.5 along the rat assumption axis. These values correspond to the true values of these parameters.

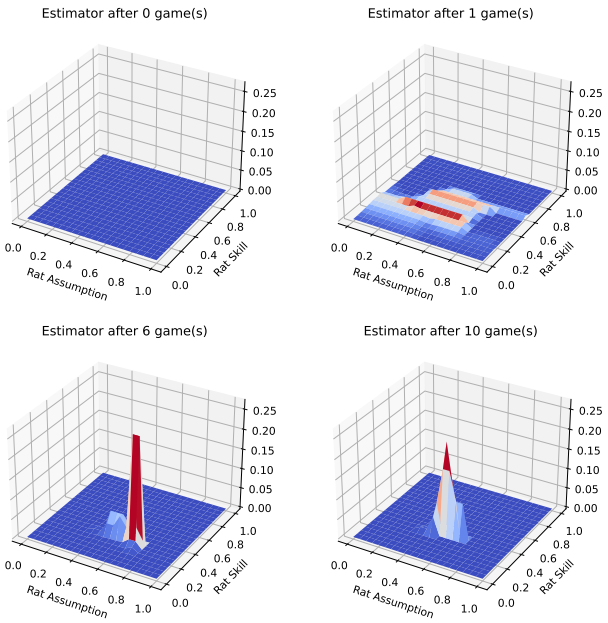


Figure A.5: Posterior distribution of cat estimating rat in the random large game.

In Figure A.6 we observe that the performance of the cat approaches the performance of the Oracle player as the skill estimate converges.

In Figure A.7 we notice that after observing 10 games the rat has been able to localise the skill of the cat accurately but not the cat's assumption about the rat's skill.

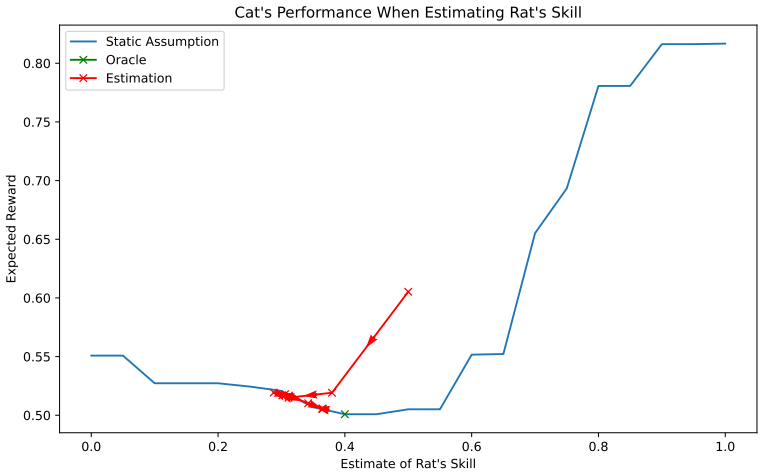


Figure A.6: Expected reward of cat estimating rat in the random large game.

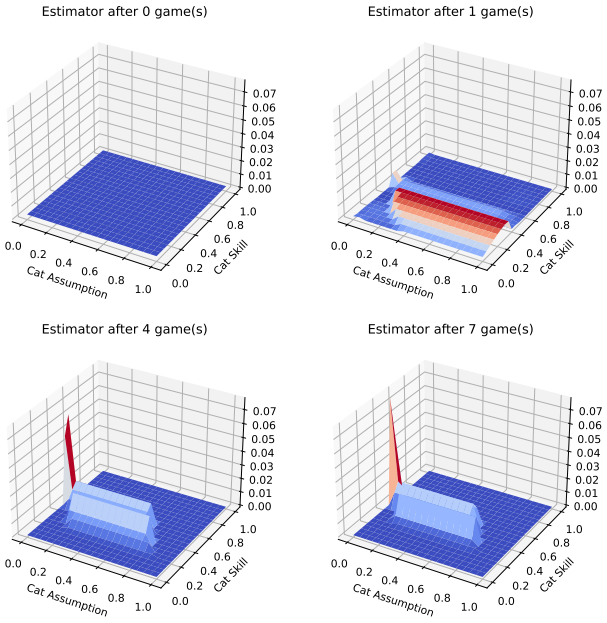


Figure A.7: Posterior distribution of rat estimating cat in the random large game.

In Figure A.8 we observe that the performance of the rat approaches the performance of the Oracle player as the skill estimate converges.

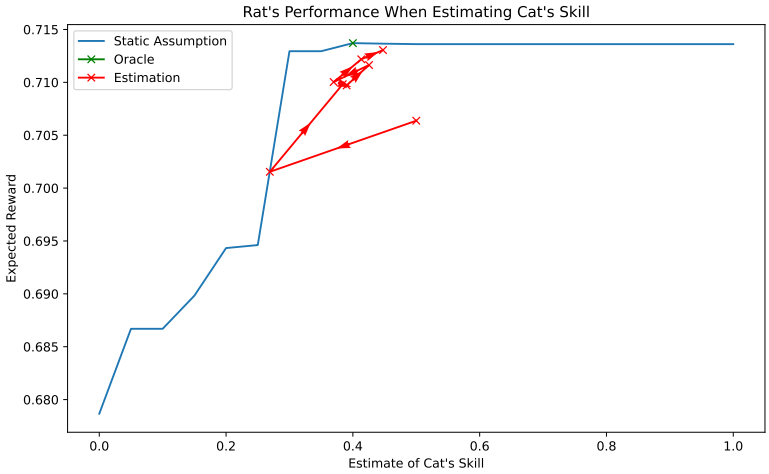


Figure A.8: Expected reward of rat estimating cat in the random large game.

A.2 Estimating Skills with Increased Failure Factor Continued

In this section we provide the relevant graphs of the players' skill estimation and performance on the random small game and random large game where the failure factor is chosen to be $\kappa = 1$. Similar graphs are presented for the contrived game in Section 6.2.

In Figure A.9 we notice that after observing 10 games the cat has been able to localise the skill of the rat accurately but not the assumption of the rat about the cat's skill.

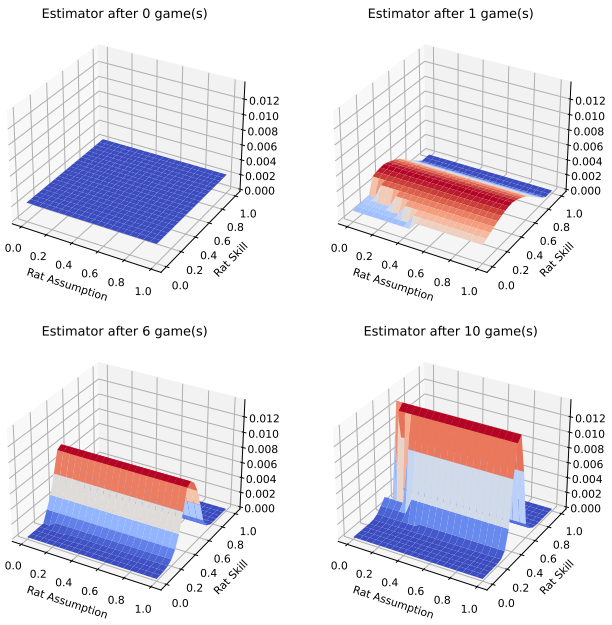


Figure A.9: Posterior distribution of cat estimating rat in the random small game with $\kappa = 0.01$.

In Figure A.10 we observe that the performance of the cat approaches the performance of the Oracle player as the skill estimate converges. We do notice that an a lucky tenth game for the rat resulted in the cat's estimate of the rat's skill being thrown off and the cat's performance worsening. The cat still performs better than it would following a uniform mixed policy.

In Figure A.11 we notice that after observing 10 games the rat has again not been able to localise the skill of the cat or the assumption of the cat about the rat's skill very accurately. As before in Figure A.3, the cat had good luck in the first 4 games and thereafter provided the rat with no further information about its skill.

In Figure A.12 we observe that the performance of the rat is worse after 10 games than a player using the uniform mixed policy. Again, like in Figure A.4 this is attributed to the cat having particularly good luck in the first 4 games and thereafter the rat gained no further useful information about the cat's skill. In Figure A.13 we notice that after observing 10 games the cat has been able to localise the skill of the rat and the rat's assumption about the cat's skill very accurately.

In Figure A.14 we observe that the performance of the cat approaches the performance of the Oracle player as the skill estimate converges.

In Figure A.15 we notice that after observing 10 games the rat has been able to localise the skill

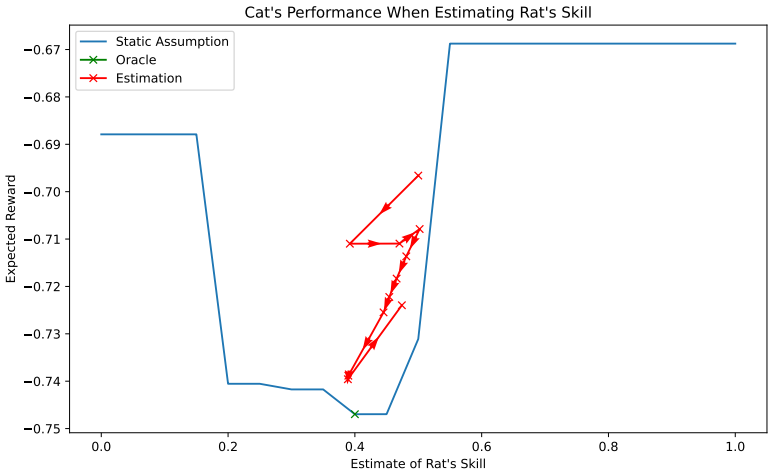


Figure A.10: Expected reward of cat estimating rat in the random small game with $\kappa = 0.01$.

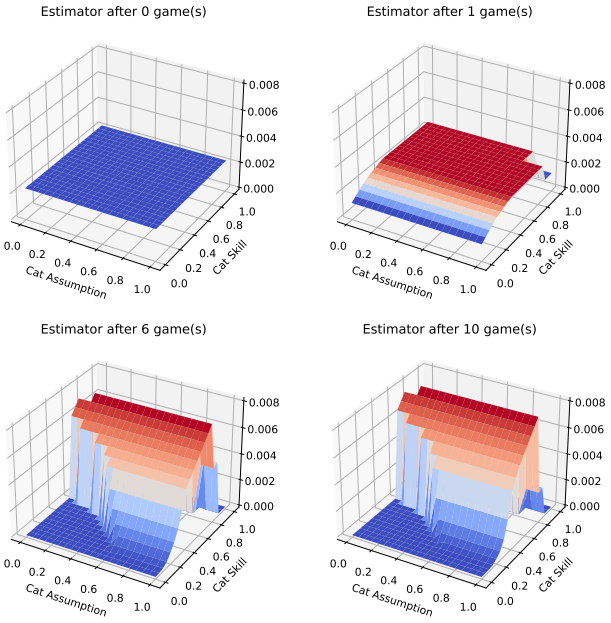


Figure A.11: Posterior distribution of rat estimating cat in the random small game with $\kappa = 0.01$.

of the cat accurately but not the cat's assumption about the rat's skill.

In Figure A.16 we observe that the performance of the rat approaches the performance of the Oracle player as the skill estimate converges.

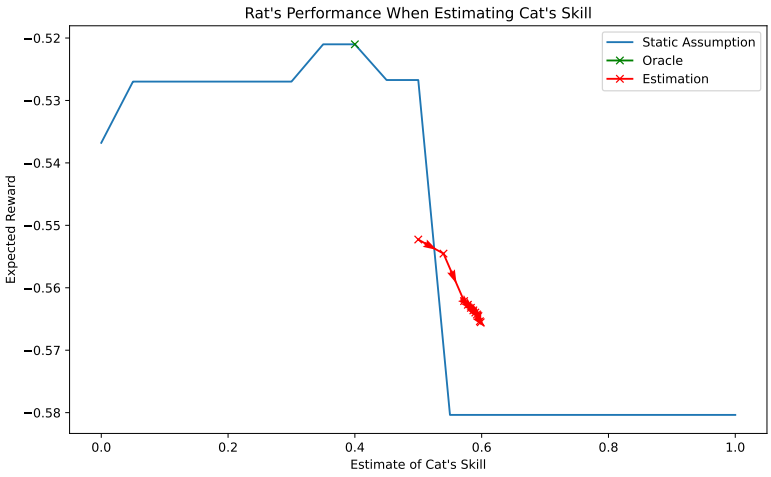


Figure A.12: Expected reward of rat estimating cat in the random small game with $\kappa = 0.01$.

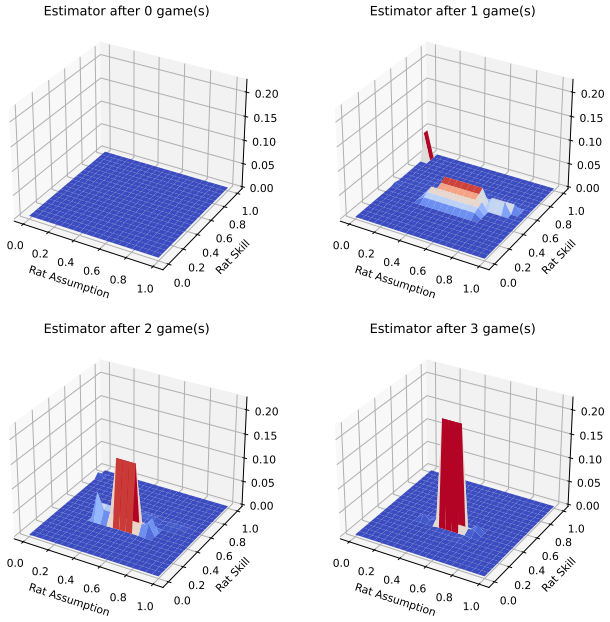


Figure A.13: Posterior distribution of cat estimating rat in the random large game with $\kappa = 0.01$.

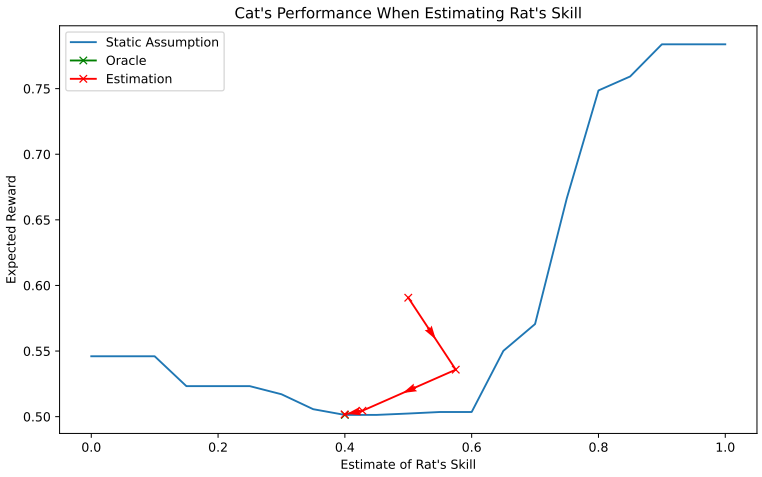


Figure A.14: Expected reward of cat estimating rat in the random large game with $\kappa = 0.01$.

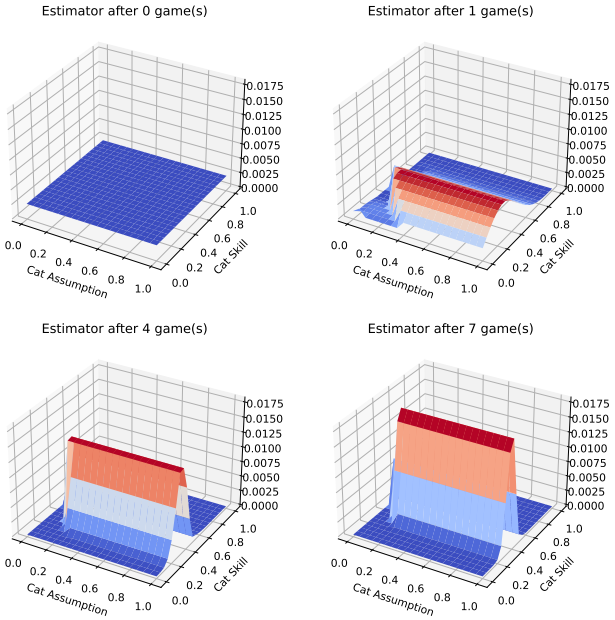


Figure A.15: Posterior distribution of rat estimating cat in the random large game with $\kappa = 0.01$.

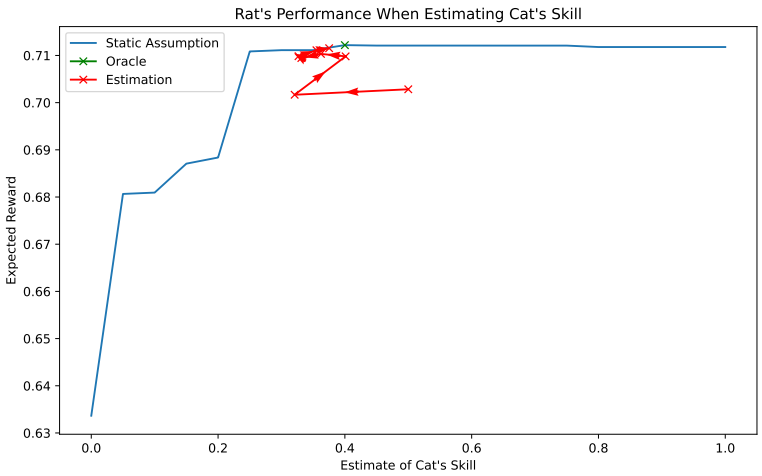


Figure A.16: Expected reward of rat estimating cat in the random large game with $\kappa = 0.01$.

A.3 Estimating the Skill of Random Players Continued

In this section we provide the relevant graphs of the players' skill estimation and performance on the random small game and random large game where the opponents follow a random policy when being observed. Similar graphs are presented for the contrived game in Section 6.3.

In Figure A.17 we notice that after observing 10 games the cat has not been able to localise the skill of the rat or the assumption of the rat about the cat's skill accurately.

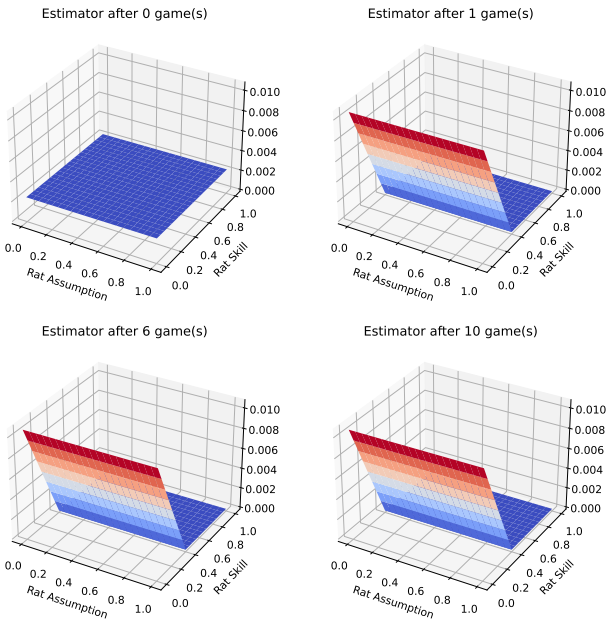


Figure A.17: Posterior distribution of cat estimating random rat in small game.

In Figure A.18 we observe that the performance of the cat does improve slightly over the uniform mixed policy, but does not approach the performance of the Oracle player.

In Figure A.19 we notice that after observing 10 games the rat has not been able to localise the skill of the cat or the assumption of the cat about the rat's skill at all and has a posterior distribution equal to the prior distribution. The cat has managed to not give away any information about its skill to the rat.

In Figure A.20 we observe that the performance of the rat does not differ from the performance of the uniform mixed policy.

In Figure A.21 we notice that after observing 10 games the cat has not been able to localise the skill of the rat or the assumption of the rat about the cat's skill accurately.

In Figure A.22 we observe that the performance of the cat does improve slightly over the uniform mixed policy, but does not approach the performance of the Oracle player.

In Figure A.23 we notice that after observing 10 games the rat has not been able to localise the skill of the cat or the assumption of the cat about the rat's skill at all and has a posterior distribution equal to the prior distribution. The cat has managed to not give away any information about its skill to the rat.

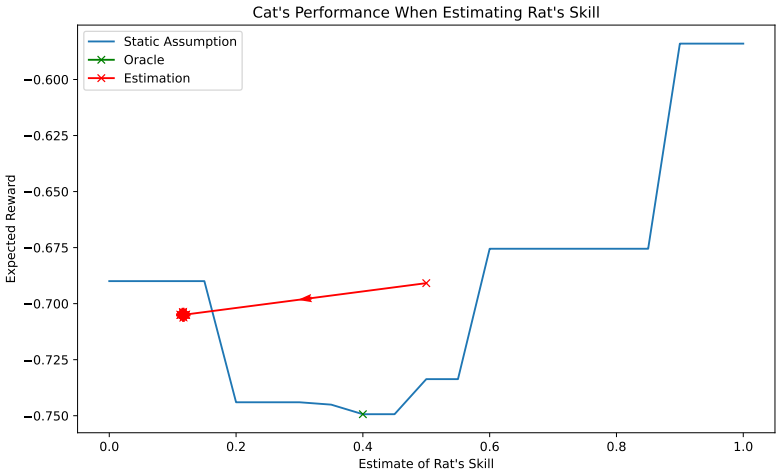


Figure A.18: Expected reward of cat estimating random rat in small game.

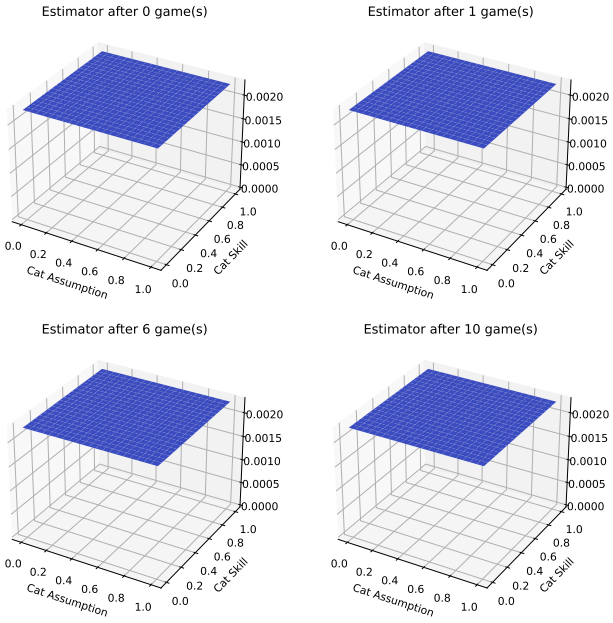


Figure A.19: Posterior distribution of rat estimating random cat in small game.

In Figure A.24 we observe that the performance of the rat does not differ from the performance of the uniform mixed policy.

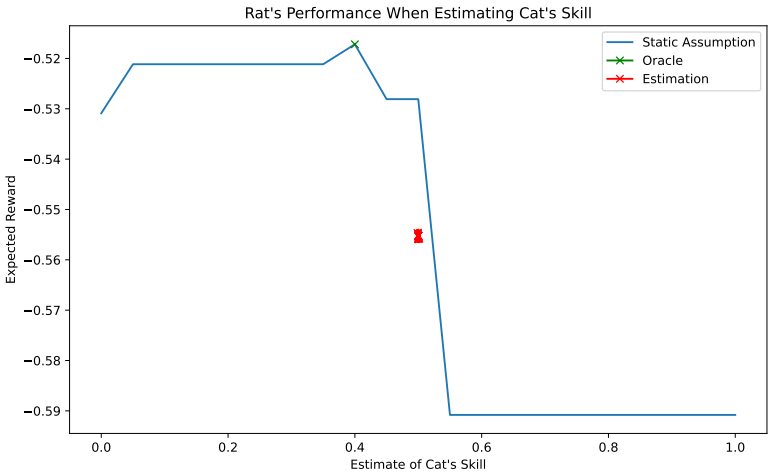


Figure A.20: Expected reward of rat estimating random cat in small game.

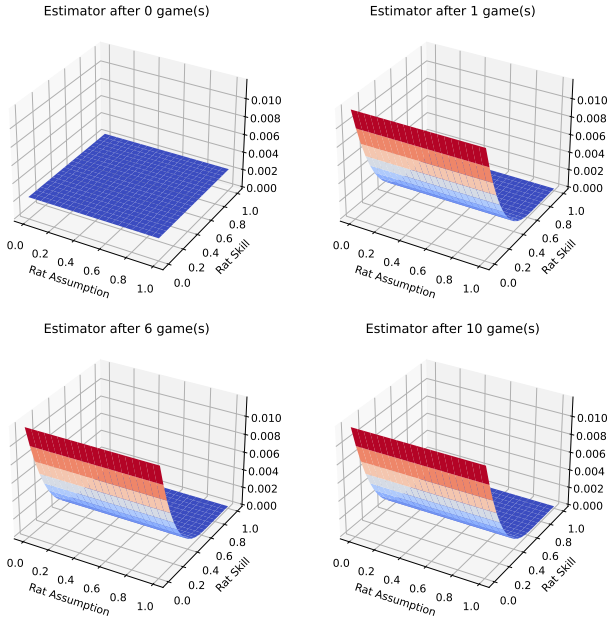


Figure A.21: Posterior distribution of cat estimating random rat in large game.

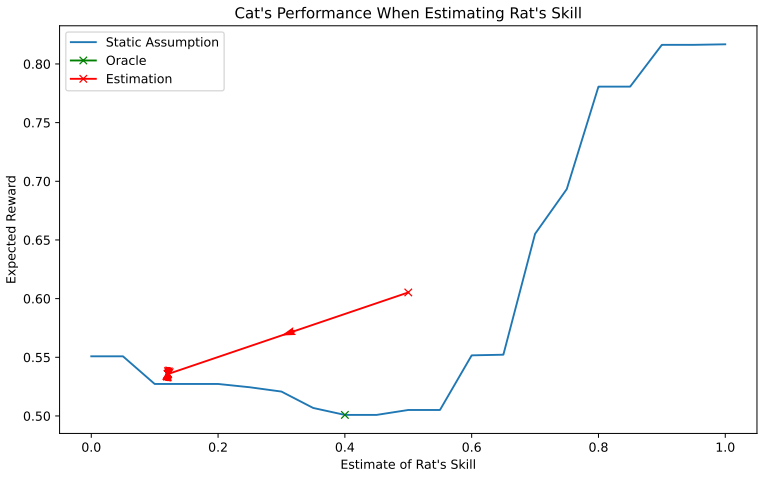


Figure A.22: Expected reward of cat estimating random rat in large game.

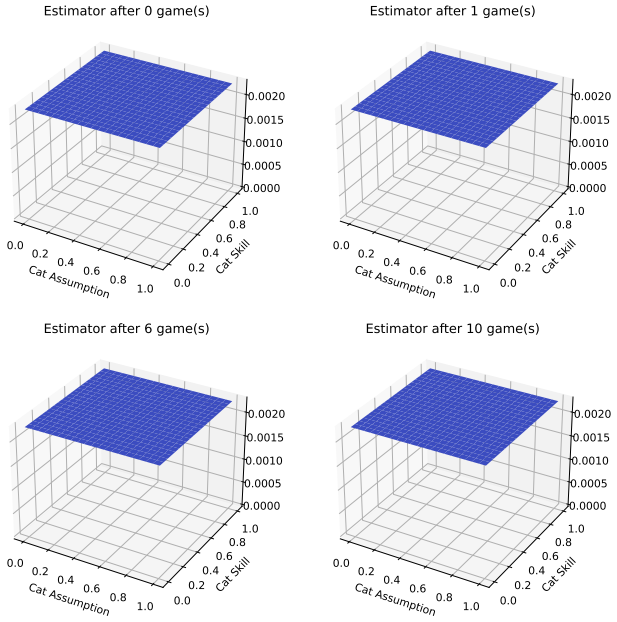


Figure A.23: Posterior distribution of rat estimating random cat in large game.

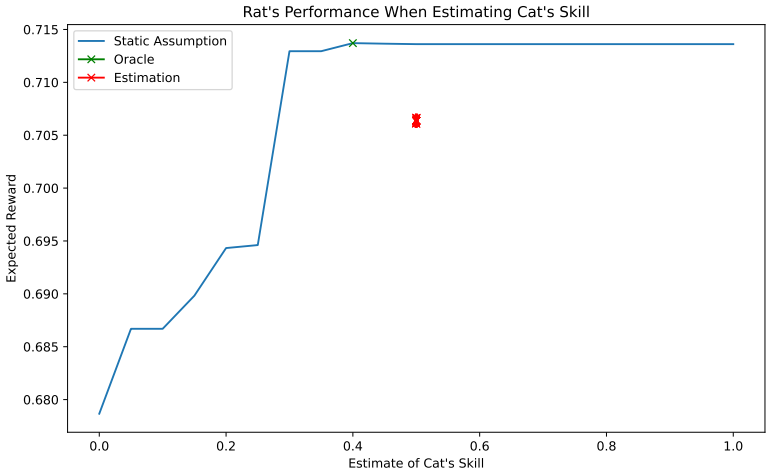


Figure A.24: Expected reward of rat estimating random cat in large game.

Bibliography

[1] John Von Neumann and Oskar Morgenstern. *Theory of games and economic behavior*. Princeton University Press, 1944.

[2] John F. Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951.

[3] Murray Campbell, A. Joseph Hoane Jr., and Feng-hsiung Hsu. Deep blue. *Artificial Intelligence*, 134:57–83, 2002.

[4] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, Jan 2016.

[5] Kevin Leyton-Brown and Yoav Shoham. *Essentials of Game Theory*. Springer, 2008.

[6] Christopher Archibald, Alon Altman, Michael Greenspan, and Yoav Shoham. Computational pool: A new challenge for game theory pragmatics. *AI Magazine*, 31(4):33–41, Dec. 2010.

[7] Christopher Archibald and Delma Nieves-Rivera. Bayesian execution skill estimation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6014–6021, Jul. 2019.

[8] Daniel Hernandez. *Opponent awareness at all levels of the multiagent reinforcement learning stack*. University of York Department of Computer Science, 2022.

[9] Samer Nashed and Shlomo Zilberstein. A survey of opponent modeling in adversarial domains. *Journal of Artificial Intellicence Research*, 73:277–327, 2022.

[10] Wan K., Wu D., Zhai Y., Li B., Gao X., and Hu Z. An improved approach towards multi-agent pursuit-evasion game decision-making using deep reinforcement learning. *Entropy (Basel)*, 23(11):1433, 2021.

[11] Dan Shen, Haibin Ling, Khanh Pham, Erik Blasch, and Genshe Chen. Computer vision and pursuit–evasion game theoretical controls for ground robots. *Advances in Mechanical Engineering*, 11(8):1687814019872911, 2019.

[12] F. R. K. Chung, Joel E. Cohen, and R. L. Graham. Pursuit—evasion games on graphs. *Journal of Graph Theory*, 12(2):159–167, 1988.

[13] Lloyd Shapley. Stochastic games. *Proceedings of the National Academy of Sciences of the United States of America*, 39(10):1095 –1100, 1953.

[14] Michael Bowling and Manuela Veloso. *An Analysis of Stochastic Game Theory for Multiagent Reinforcement Learning*. Carnegie-Mellon University Pittsburgh PA School Of Computer Science, 2000.

[15] Hyeong Soo Chang. Value set iteration for two-person zero-sum markov games. *Automatica*, 76:61–64, 2017.

[16] Richard Bellman. A markovian decision process. *Journal of Mathematics and Mechanics*, 6(5):679–684, 1957.

[17] A. Wrobel. On markovian decision models with a finite skeleton. *Zeitschrift für Operations Research*, 28:17–27, 1984.

[18] Donald Michie. Game-playing and game-learning automata. In *Advances in Programming and Non-Numerical Computation*, pages 183–200, 1966.

[19] Robert V. Hogg, Elliot A. Tanis, and Dale L. Zimmerman. *Probability and Statistical Inference*. Pearson, 10 edition, 2021.

[20] Luis von Ahn. Preliminaries of game theory. `https://web.archive.org/web/20111018035629/http://scienceoftheweb.org/15-396/lectures_f11/lecture09.pdf/`. 2011-10-18.

[21] Eric Van Damme. Chapter 41 strategic equilibrium. In *Handbook of Game Theory with Economic Applications*, volume 3, pages 1521–1596. Elsevier, 2002.

[22] Reinhard Selten. Spieltheoretische behandlung eines oligopolmodells mit nachfrageträgheit. *Zeitschrift Für Die Gesamte Staatswissenschaft*, 121:301–324, 1965.

[23] Reinhard Selten. The chain store paradox. *Theory and Decision*, 9(2):127–159, 1978.

[24] R. Selten. Reexamination of the perfectness concept for equilibrium points in extensive games. *International Journal of Game Theory*, 4:25–55, 1975.

[25] David Kreps and Robert Wilson. Sequential equilibria. *Econometrica*, 50(4):863–94, 1982.