Investigating the Models People Use to Guide Decision-Making Behaviour

An Analysis of Human Performance on an Optimal Experimental Design

Patrick Laverty

THE UNIVERSITY *of* EDINBURGH

MInf Project (Part 2) Report

Master of Informatics

School of Informatics

University of Edinburgh

2020

Acknowledgements

First and foremost, I would like to say thank you to my supervisor, Chris Lucas, for his advice, ideas, and guidance over the past two years of this project. Initial conversations with him lead to the theory that was developed and tested in this project. I will be forever grateful to him for inspiring in me a deep curiosity and love for cognitive science, a field I hope to have a long a fruitful career in.

My thanks also go out to Michael Lee for helping me understand the models I have used in my analysis, as well as how to best interpret the results presented in Chapter 5. His enthusiasm for this project inspired me to work on it to the best of my ability.

Thank you also to Rosa Filgueira, who, unknowingly, helped me understand how to get Spark to run on the Eddie computer cluster by creating a tutorial on the subject. Without this tutorial, the analyses conducted in this project would be severely limited.

Thank you to my friends, old and new, particularly Evan and crew for their positivity in these strange and uncertain times.

Finally, I would like to thank my parents for their unconditional love and support, not only throughout my university career but throughout my entire life. I would not be where I am today if it weren't for them.

Abstract

Results have shown a propensity for individuals to use the Extended Win-Stay Lose-Shift (e-WSLS) decision-making strategy when solving bandit problems. This model has no concept of the interval in which problems are solved, nor do they use memory to integrate information on arms based on previous trials. Both factors lead to decisions made under the model which are rash and sub-optimal. The theory proposed by this project is that the e-WSLS model is the dominant strategy people use when solving bandit problems. Design optimization (DO) is a technique in which the criteria for experimental design are quantified and optimized, resulting in an optimal experimental design (OED) which researchers believe is more informative than a naïve design. OEDs allow for more robust testing, and possibly the need for fewer experimental samples to draw robust conclusions. A DO framework was developed for this project, with the goal of producing a design where the e-WSLS model performed poorly relative to other models, and where it was maximally differentiated from those other strategies. Those criteria were chosen so that the design produced would induce participants in an experiment to use a model other than the e-WSLS model, thus falsifying the claim that it is a dominant decision-making strategy. Findings show that the DO framework was successful in both differentiating the e-WSLS model from all others, and in producing a design where the e-WSLS performed sub-optimally relative to other models. The OED

was used in an experiment with 138 participants, and an analysis using hierarchical Bayesian models found that while the proportion of participants using the e-WSLS strategy dropped under the OED when compared to a naïve design, the model still accounted for the most significant proportion of participants in the dataset.

Contents

Chapter 1

Introduction

Imagine yourself arriving at your favorite restaurant, one you have visited since before you can remember. You are greeted at the door by your favorite server who always greats you warmly and seats you at your favorite table. Today, however, they seem off: their greeting is abrupt, and they seat you not at your favorite table, but one near the front door where a draft comes in every time a new patron arrives. You ignore this, however, instead focusing your attention on what you are going to order! Of course, you know that you will order your regular: a tried and true recipe that has never disappointed. When your food arrives, however, it is not what you expected it to be: the portion doesn't seem as large as it usually does, and something is off in the flavor - perhaps an ingredient is missing, or the food was slightly overcooked. When it's time to pay and leave, you notice the bill is slightly higher than you expected; you later notice that an extra item was erroneously added to your bill. All-in-all, this has been a bad experience at a typically outstanding restaurant. You have visited this restaurant for years and have never once had an unpleasant experience. You have also tried all of the other restaurants in the area and found, much to your dismay,

that none compare to this particular favorite of yours. You decide, however, that you won't return to this restaurant again - not at least until you go back and try all of the others which have only disappointed in the past.

The type of decision-making strategy used to arrive at the conclusion above is reminiscent of the Win-Stay Lose-Shift strategy. With Win-Stay Lose-Shift, you stick with an option as long as it continues to generate rewards, but at the first sign that rewards have stopped, you switch to an alternative. In the example above, the reward is an enjoyable dining experience, and the alternative is any other restaurant, regardless of whether the other restaurant has only disappointed in the past. As evidenced by the questionable decision not to return to your favorite restaurant after one bad experience, decisions made under the Win-Stay Lose-Shift strategy may seem a bit rash. Evidence, however, suggests that this strategy is employed by humans in a wide variety of situations (Nowak & Sigmund, 1993; Vickery, Chun, & Lee, 2011; Worthy, Hawthorne, & Otto, 2013; Christian & Griffiths, 2016).

Win-Stay Lose-Shift (WSLS) was first introduced in 1952 by Herbert Robbins as a heuristic strategy for choosing between 2 decisions in a sequential sampling task, providing a proof that as the number of samples drawn becomes very large, WSLS is better than choosing at random (Robbins, 1952). It was later applied to the prisoners dilemma[1] and was found to dominate the Tit-for-Tat strategy - this enormously successful strategy involves copying what your opponent did to you on the previous turn

[1]For those unfamiliar with the prisoner's dilemma, it is a classic game often analyzed in game theory. The canonical example includes two players - prisoners believed to have committed a crime and are now being interrogated - who each have two options: cooperate or defect. Each player must choose to cooperate or defect, without knowing what the other player does. The outcome of a round depends on the choice made by both players.

(Nowak & Sigmund, 1993). Nowak and Sigmund (1993) note that WSLS - dubbed the Pavlov strategy in the original paper - may work in contexts other than prisoner's dilemma. The authors also note that more sophisticated variants may exist in which memory of the previous few rounds may be used to shape decisions, rather than only the previous round. Models involving memory, however, are often classed as fundamentally different than the WSLS model, for example, the ϵ-greedy and ϵ-first models which encode past rewards and (Sutton, Barto, et al., 1998). Others have refuted the claim that WSLS would work in contexts other than the prisoner's dilemma (Ivan, Banks, Goodfellow, & Gruber, 2018). The WSLS is often criticized for its lack of memory (Ivan et al., 2018; Gutierrez et al., n.d.; Christian & Griffiths, 2016), as well as inattention to the horizon, noting that WSLS denotes you should use the same kind of decision on your first trial as you should on your last (Christian & Griffiths, 2016).

Regardless of the models' simplicity, many studies have been conducted on human decision-making which have found prevalent use of the strategy across many domains of human decision, including gambling tasks (Worthy et al., 2013) and studies of social dilemma games (Nowak & Sigmund, 1993; Vickery et al., 2011); as well as decision making in monkeys (D. Lee, Conroy, McGreevy, & Barraclough, 2004).

Last year's project involved investigating which models of decision-making people used when solving bandit problems (Laverty, 2019). As such, of interest to this project are studies that tested the use of WSLS in bandit problems. The WSLS model - or a variant of it which allows for different probabilities of staying with a win and shifting after a loss dubbed the Extended Win-Stay Lose-Shift strategy (e-WSLS) - has been found to account for human behavior in many experiments involving bandit problems

(Steyvers, Lee, & Wagenmakers, 2009; Guan, Stokes, Vandekerckhove, & Lee, n.d.). The finding was further corroborated by the results of last year's project, which found that 62% of participants who solved a series of bandit problems used the e-WSLS model, with a further 6% using the simpler WSLS variant (Laverty, 2019).

The use of WSLS is not only found in laboratory experiments: it has also been detected when analyzing decision-making strategies of those in managerial positions (Tamura & Masuda, 2015).

The model isn't only sub-optimal in hypothetical thought experiments: Ivan et al. (2018) found that those under cognitive strain defaulted to a lose-shift dominant strategy, the same strategy used by children aged 5-9, but not other adults who had no additional cognitive load placed on them. In their experiments, a random strategy would have achieved a higher score than WSLS. The researchers in the aforementioned study regarding WSLS use in those in managerial positions found that decisions made under this strategy lead to worse performance outcomes (Tamura & Masuda, 2015). A study also found that WSLS had the lowest agreement with optimal data compared to other cognitive heuristic models (M. D. Lee, Zhang, Munro, & Steyvers, 2011). The prevalence of this model, coupled with its simplicity, may be evidence that it is a fast and frugal cognitive strategy like those proposed by Gigerenzer et al. (Todd & Gigerenzer, 2000). Some researchers have explicitly stated that WSLS, particularly the lose-shift aspect, is "a default and reflexive response strategy in humans that is normally suppressed by executive functions" (Ivan et al., 2018).

The results that we use the same decision-making strategy as five-year-old children and monkeys aren't promising. With both experimental results and real-world examples showing a prevalence of this simple, sub-optimal, model it's essential to

understand where it is used and to what extent. Understanding model use is an important goal in modeling in the cognitive sciences. The importance of knowing what model people use is that we can attempt to teach them better strategies (M. D. Lee et al., 2011; Laverty, 2019).

One question that is often ignored is that of how reliable the results mentioned above are? While one paper found the e-WSLS model had the highest agreement with human data (Steyvers et al., 2009), another found it had the lowest (M. D. Lee et al., 2011). What is to be made of these results? Before drawing any conclusions about a particular theory, therefore, it is vital that we are able to test it in a robust manner. To do this, we can turn to results obtained from experiments derived from design optimization frameworks.

Design optimization (DO) is a framework which quantifies the process of finding an optimal experimental design, turning it into a mathematical problem which can be optimized (Sun, 2012). DO has been studied across many fields including statistics (Fisher, 1936), economics (El-Gamal & Palfrey, 1996), machine learning (Settles, 2009), and psychology and cognitive science (Myung & Pitt, 2009; Zhang & Lee, 2010b).

By using the DO framework, designs that are more informative than their naïve counterparts are produced, allowing for more robust conclusions from experiments using such designs to be drawn (Heck & Erdfelder, 2019). The question of what "informative" means is relative to the study at hand, with many experiments in cognitive psychology optimizing for maximum differentiation between competing models (Myung & Pitt, 2009; Heck & Erdfelder, 2019). The thought process here is that there are often many models that could describe a possible dataset, and so designs

in which these models make qualitatively different predictions should be preferred (Myung & Pitt, 2009). Myung and Pitt (2009) provide a framework for DO which enables researchers to find an optimal experimental design (OED) that maximally differentiates between competing models by maximizing a utility function, tailored to the specific cognitive models being investigated. Zhang and Lee (2010b) later adapt it to discriminate between pairs of models on Bandit designs.

In a survey of the applications of DO, the author highlights many examples where the framework was used in experimental designs in psychology (Sun, 2012). Of all of the experiments considered, however, model discrimination was the only criterion by which designs were optimized. While this type of optimization is useful for better discrimination between competing models, it is of limited use for experiments that seek to falsify a specific theory.

Karl Popper, a renowned Philosopher of Science, remarked that science seeks to disconfirm, and that confirmations should only count if they come from risky predictions, i.e., those that seek to destroy a theory (Popper, 2014). He stated that the only proper test of a theory is one that is trying to falsify it (Popper, 2014). Relating these claims to the domain of cognitive science, others have noted that in ordered to test rigorously test a cognitive model, experimental designs must be optimized in order to increase the chances of falsification (Heck & Erdfelder, 2019). A famous paper by Roberts and Pashler (2000) states that significant evidence can only be provided for a theory when both the theory and the data used place substantial constraints on the possible outcomes of that theory; the degree to which the theory and data provide such constraints is dependant on the experimental design used (Heck & Erdfelder, 2019).

As stated already, many results suggest that the WSLS model, or a variant thereof, is a dominant model of human decision-making in many domains. Of interest to this project is whether the (e-)WSLS model is the dominant strategy used by humans when solving bandit problems, with results suggesting that this may be the case (Steyvers et al., 2009; Zhang & Lee, 2010a; Laverty, 2019). To test this theory, following the test of falsifiability proposed by Popper, I shall attempt to falsify it. I believe that one way to falsify this theory would be to use an experimental design in which people should use a decision making strategy other than the (e-)WSLS model. Under this design, we should see a marked decrease in the proportion of participants of an experiment using the (e-)WSLS model. If we do, this would falsify the theory that it is the dominant decision-making strategy used by humans when solving the bandit problem. If we do not, then additional credence should be given to the original theory.

To increase the chances of falsification, I shall, therefore, use an experimental design produced by a design optimization framework. As noted already, current DO frameworks aim to maximally differentiate between models believed to describe a set of data. This criterion will be useful when analyzing the results of the experiment when identifying the model with the largest agreement with participants' data. In order to test my theory, however, I shall have to extend such a framework to produce an experimental design where participants should *not* use the (e-)WSLS model. An overview of the literature of the field suggests that there has been no previous attempt to adapt DO frameworks to such a task; this extension is a novel contribution of this project.

Structure of this report

This report will proceed as follows: in Chapter 2 I will give a summary of the work conducted in the first year of this project. I will relate this to the work that was carried out this year, as well as provide the concrete goals I sought to achieve in this project. In this chapter, I will also provide brief definitions of key terms and ideas which will frequently arise throughout this report. Chapter 3 will provide readers with descriptions of the models used in this project. In Chapter 4, I will describe the design optimization framework I have developed to test the theory outlined above. This chapter will include a brief analysis of the output of that DO framework, as well as a description of an experiment conducted using an optimal experimental design produced by this framework. Chapter 5 will then detail the results of this experiment, along with an analysis of the models described in chapter 3 on both human and optimal performance on the task. Finally, Chapter 6 closes this report with a few remarks on what was achieved throughout this project, as well as some potential avenues for future research.

Chapter 2

Previous Work

2.1 Background

In this section, I wish to give readers the context for this year's work by describing briefly the work that was carried out in the first half of this project. Doing so will require giving definitions of key terms that will be used frequently, as well as descriptions of tasks and models which are the subject of study in this project. The significant part of the report for the first year of this project was a literature review, focusing on the five key areas relevant to this project: the exploration-exploitation tradeoff; bandit problems; mathematical models of human cognition; the study of individual differences; and hierarchical Bayesian models. I do not believe I am able to give a better account of those areas than I did in last year's report, nor would it be worthwhile to attempt to do so. As such, I have included, verbatim, the descriptions of each of those five areas from last year's project. All work in this background section should be assumed to have come from last year's report, and credited as so. The only difference between the two sections in both reports is that I have attempted

to significantly shorten the descriptions, including only what is relevant to this year's project. This should not be seen as an attempt to pass off work conducted last year as new, simply because I have edited it; the editing has taken place only so that this section is not overly long and filled with unnecessary information. For those interested in a more in-depth discussion of any topics mentioned in this section, I refer you to Chapter 2 of last years report (Laverty, 2019).

2.1.1 Exploration-Exploitation Tradeoff

Initial work on this project was spurred on by investigations into how humans solve the *exploration-exploitation dilemma* (Laverty, 2019). For those unfamiliar with the term, the exploration-exploitation dilemma refers to the problem that arises when we are tasked with making decisions in unknown environments. Two options that arise are: to query many different options in order to gather more information on what option may be best (i.e., explore); or to make use of the information we have and choose the best option encountered so far (i.e., exploit). The dilemma arises when we try to determine how we should balance these two options. We can see that both strategies come with downsides: explore too much, and you end up making many sub-optimal decisions, exploit too much, and you may get stuck in a local optimum. Of interest to many researchers is how humans balance the need for both exploration and exploitation (Sutton et al., 1998; Aston-Jones & Cohen, 2005; Hills et al., 2015).

In order to investigate this phenomenon, I turned to research on a common task used to investigate how humans navigate the exploration-exploitation tradeoff: bandit problems.

2.1.2 Bandit Problems

In bandit problems, an agent is presented with several options known as bandits[1], and are told to "play" the bandits to maximize their reward over some amount of time or number of trials. In the classic set-up, an agent selects one bandit per time step and receives a reward according to a Bernoulli process, parameterized by some payout rate, θ, for each arm, which is unknown to the agent at the beginning of the task. The payout rate differs for each arm, and it is expected that the agent will generate an estimate of each arms payout rate through repeated plays.

Many variants of this original structure exist, each motivated by the desire to capture more complex and interesting real-world phenomena, including: bandits with delayed rewards; bandits where contextual information is used; and non-stationary bandits with changing reward rates.

It is important here to highlight these variants, as different experimental set-ups have different optimal solutions, and models applied to one task are not appropriate for study in another. I should, therefore, clarify at this stage that the bandits that are the subject of my study and analysis are: 4-armed and 5-armed Bernoulli bandits, both played with a finite horizon and a stationary reward policy. These designs were chosen both for their simplicity of analysis and availability of data.

Bandit problems have been used to study the exploration-exploitation tradeoff in a wide variety of domains (Steyvers et al., 2009; Sutton et al., 1998; Shen, Wang, Jiang, & Zha, 2015). As such, many models have been proposed as solutions to bandit problems all of kinds (Berry & Fristedt, 1985; Burtini, Loeppky, & Lawrence, 2015). In the next section of this report, I will describe a type of model based on

[1]Named for their similarity to slot-machines, commonly referred to as bandits.

psychologically inspired heuristics that people are believed to use when solving bandit problems.

2.1.3 Heuristic Cognitive Models

For cognitive scientists, bandit problems allow conclusions to be drawn about how the human mind deals with uncertainty, balancing exploration and exploitation, and the reward policies under which humans are operating. In order to investigate these questions, quantitative mathematical models with explicit psychological content are used[2]. In such models, values of psychological variables can be analyzed in order to make inferences on how humans solve a task. The models can also be applied to both human and optimal data in order to examine the relationship between human and optimal performance. Doing so also allows researchers to determine which models give the best account of optimal data, and thus could be used as a computationally tractable heuristic - the method of generating optimal decision-making data via dynamic programming methods is computationally expensive, as will be discussed in Chapter 3. In addition to this, we can also determine where human performance is falling short of optimal and use this information to teach decision-makers.

The details of the specific models under consideration are left to Chapter 3 of this report. It is sufficient to say now that various models have given a good account of both human and optimal behavior, and analysis of the psychological variables involved have provided insights into how humans manage the exploration-exploitation tradeoff.

[2]commonly referred to in the literature as process models (Farrell & Lewandowsky, 2018)

2.1.4 Individual Differences

The study of individual differences is a field in-and-of-itself in the broader category of the study of psychometrics, with the study of individual differences in cognition dating back to 1973 (Hunt, Frost, & Lunneborg, 1973). While they have been studied in many fields, little attention has been paid to individual differences in the cognitive sciences (Zeigenfuse & Lee, 2009). Farrell and Lewandowsky (2018) report that "[m]ost psychological experiments report data at the group level, usually after averaging the responses from many subjects in a condition". The benefits of aggregating data include the fact that it is simpler to perform an analysis on one averaged data set than many individual ones, both in terms of human and computational resources, and the risk of overfitting on data is minimized. However, Farrell and Lewandowsky (2018) warn that "averaging may create a strikingly misleading picture of what is happening in [an] experiment". In particular, when working with human data, aggregation assumes that all individuals are the same and ignores the complexities of human psychology and the multitude of factors, and their interactions, which come into play during decision-making. In the past, research has been redacted, and different conclusions have been drawn, based on whether models were fit on aggregated or individual data (Heathcote, Brown, & Mewhort, 2000).

While it is impossible when developing cognitive process models to account for the infinite variations in factors acting upon an individual, the very basic idea that individual variation is to be expected should be accounted for. When dealing with bandit problems, we are attempting to make inferences about how humans manage the exploration-exploitation tradeoff; such inferences would be unfounded if we don't account for the fundamental fact that individuals vary in their biases towards explo-

ration and exploitation.

While aggregating data has its disadvantages, simply fitting models to the data of each individual does not allow for meaningful analyses either. We want a way of developing models that are able to best fit individual participants, while also allowing inferences to be made about humans in general, and on how different individuals, and groups of individuals, compare to each other in how they solve similar problems. Frameworks for conducting such analyses are described in the next section of this report.

2.1.5 Hierarchical Bayesian Models

In a special issue of the *Journal of Mathematical Psychology*, all articles presented applied HBMs to previously studied tasks in cognition, highlighting the flexibility of Hierarchical Bayesian Models (HBMs), as well as the benefits they provide to researchers M. D. Lee (2011b). These benefits include: allowing researchers to form deeper theories with richer psychological content; allowing the same set of parameters to be used to explain behavior across different but related tasks; and allowing for fundamentally different models to be mixed and unified to explain oberserved data better. This section will detail extensions to simple models of cognition that have been proposed in the past.

Hierarchical Modelling

Cognitive process models, in their simplest form, assume that data, d, is generated by some function - a model - which is parameterized by some set of parameters, θ. A graphical representation of this idea is shown in Figure 2.1.

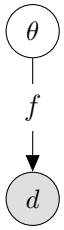


Figure 2.1: Non-hierarchical cognitive model

The exact definition of what makes a model "hierarchical" is not defined, with most researchers providing a definition by example. In order to continue with this section of the report, the definition by M. D. Lee (2011a) is adopted which states that: "we treat as hierarchical any model that is more complicated than the simplest possible type of model shown in [Figure 2.1]."

A simple and immediate extension to this model is to assume that the parameters of the generating process, f, are themselves generated by yet another process, g, with it's own set of parameters, λ; Figure 2.2 shows this extension, and makes clear the inspiration for the name "hierarchical" model.

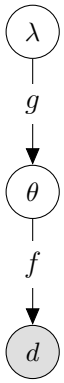


Figure 2.2: Hierarchical extension of simple cognitive model

One inspiration for the need for such models lies in the desire to accommodate individual differences. M. D. Lee (2011a) describes in his overview of HBMs that: "The non-hierarchical approach has to rely on first doing separate inference for parameters

and data for each person, and then trying to say something about individual differences through post-hoc analyses. In the hierarchical approach in [Figure 2.2], the structure in individual differences is directly captured by the process [g] and its parameters [λ]."

The kind of model presented in Figure 2.2 not only allows for individual differences "but imposes a model structure on those differences, and allows inference about parameters – like the group mean and variance – that characterize the individual differences." (M. D. Lee, 2011a).

Mixture Modelling

We might expect that different groups of individuals may solve a problem in fundamentally different ways. As such, we would want to capture this phenomenon when developing a process model. We could imagine that any number of groups may exist and that each group solves the same problem using a different approach. In this case, a mixture model would best explain the data. According to Farrell and Lewandowsky (2018) "Mixture modeling ... [applies] to cases where we suspect that different participants might perform the task differently, either due to discrete differences in ability or due to differences in strategies used, but where we have no external indicator of the subsets except for performance on our task." HBMs provide a ready framework for such models to be built. The model shown in Figure 2.3 contains multiple processes, f_1 through f_n, each with its own set of parameters, θ_i. A latent variable, z, along with a mixture process, h, is added to the model to determine how the different processes are used to generate the data: this may be a discrete process which allows for only one model to be used at a time, or it may combine results from each process, weighting them according to the latent mixture parameter - the specifics of the model will

change from task to task, however, these different options are highlighted to exemplify the flexibility of such models.

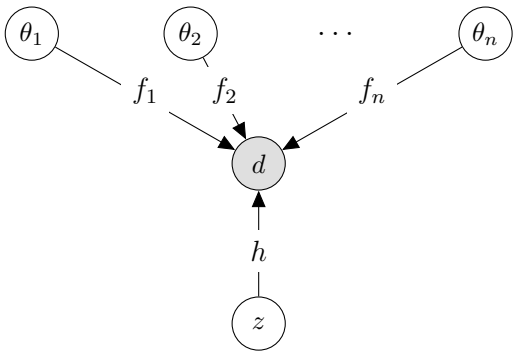


Figure 2.3: Mixture model allowing processes to combine to produce the observed data

Finally, the most ambitious type of HBM is presented in Figure 2.4.

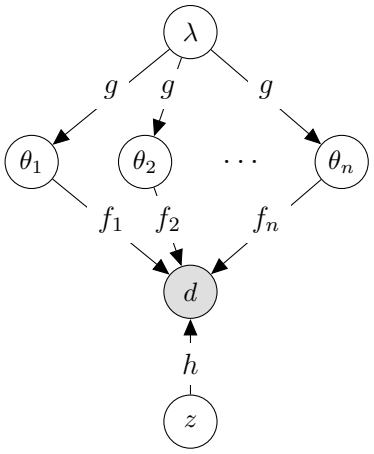


Figure 2.4: Hierarchical extension of a mixture model

This model assumes that a mixture of processes can be used to explain the observed data, with each process' parameters differing, but stemming from the same underlying distribution. Of course, we don't have to stop at just one level of hierarchical abstraction: we can - as some models detailed in Chapter 3 of this report do - encode in our model the notion that each θ_i is generated according to its own process, g_i, parameterized by some values λ_i. We can then place a prior on these

values of λ_i, assuming perhaps that they are generated by some underlying process, m, parameterized by yet another value, ϕ; or we can go further up the hierarchy, assuming individual generating processes for each λ_i, and so on.

In the next chapter, existing models that have been applied to bandit problems are described, and details are presented as to how they can be expanded via the hierarchical Bayesian framework.

2.2 Findings from Year 1

Last year's project applied previously developed heuristic models to a data set consisting of decisions made on bandit tasks by 451 participants gathered by Steyvers et al. (2009) - referred to from here on out as the *testweek* dataset[3]. Each participant in the testweek experiment completed 20 4-armed bandit problems, each of which had a different reward distribution over the 4 arms, and each problem lasting for 15 trials. The researchers who gathered the data found evidence of individual differences in terms of how subjects grappled with the exploration-exploitation tradeoff.

Other researchers have used the dataset as a testing ground for new models, with Zhang and Lee (2010a) building and testing a hierarchical Bayesian mixture model composed of four psychologically inspired heuristics models from the reinforcement learning literature. Those researchers also found that there was clear evidence of individual differences captured in the dataset.

This dataset was, therefore, chosen for study in last year's project due to it's size - it is the largest dataset of human decision-making on bandit problems - and the fact that there is multiple evidence of capturing individual differences in human performance

[3]The data was collected during testweek at the University of Amsterdam.

on bandit problems. These factors make it is the best existing dataset for testing new theories and models concerning human performance - and individual differences in human performance - on bandit problems.

The project began with a replication of work conducted in both papers mentioned above, (Steyvers et al., 2009; Zhang & Lee, 2010a), in order to test the findings of those papers concerning individual differences in model use captured by the dataset (Laverty, 2019, Chapter 4). Results concerning individual differences in model use across both studies were almost perfectly replicated (Steyvers et al., 2009), with minor differences in results attributed to differences in the implementation of the models, and confusion over priors placed on the Bayesian models of (Zhang & Lee, 2010a). These replication studies concluded that the dataset captured could be reliably used to study individual differences in model use across a wider range of models.

The next stage of the project was to conduct such an analysis. The first model I sought to test was a newly proposed model dubbed the τ-Switch model (M. D. Lee, Zhang, Munro, & Steyvers, 2009), which was found to outperform all other psychological heuristics considered when tested on a dataset of 10 individuals (M. D. Lee et al., 2009). The original τ-Switch model was developed for use on 2-armed bandit tasks, and so to test it on the 4-armed bandits of the testweek dataset I extended it for use on N-armed bandit problems.

The τ-Switch model, along with 2 other psychologically inspired heuristic models, were also added to the hierarchical Bayesian model developed by (Zhang & Lee, 2010a). The goal of developing this model was to test both within-group and between-group individual differences using the largest mixture of cognitive models tested on this dataset to date. The agreement of this model with the testweek dataset was also

calculated in order to determine whether a mixture model with a larger number of component models was able to give a better account of human decision-making data than simpler mixture models. Finally, two extensions to the hierarchical Bayesian mixture models discussed thus far were developed. These extensions involved allowing for latent switching between the model used by a subject on a per-game and per-trial basis.

All the models described above were applied to the testweek dataset, and analyses of the agreement of these models with the data, as well as a study of individual differences captured by each model, were conducted (Laverty, 2019, Chapter 5).

Results of an analysis including the τ-Switch model were contrary to previous findings where the model outperformed a suite of other psychological heuristics (M. D. Lee et al., 2009), with the τ-Switch model best describing only 2% of participants in the testweek dataset.

Analyses of the extended hierarchical Bayesian mixture model found that it gave a better account of human performance than the simpler mixture model. The same analysis showed that the mixture models which allowed for per-game and per-trial switching in model use gave a better account of the data than equivalent mixture models which limited on model per participant throughout the entire experiment. The extended hierarchical Bayesian mixture model allowing for per-trial switching - the most flexible model in the set - gave the highest agreement with human data than all other models considered. Finally, in a study of individual differences in model use, the model which gave the best account of human data found that 71% of participants were believed to be using the e-WSLS model as their primary decision-making strategy when solving bandit problems.

2.3 Errors / Corrections I wish to pursue

It was pointed out to me during email correspondence with the original developer of the τ-Switch model that the model I implemented did not capture the correct behavior of the model in the "exploit" state. The developer of the original model described the correct behavior as follows[4]:

"... we could have arm A with (S=3, F=2), B with (S=3, F = 1) and C with (S=0, F=0), right, and we'd like to pick B if we're in the exploit state. If I'm following the current specification correctly (I might not be) nothing is said about A vs. B."

The model I implemented would simply choose randomly between arms A and B when, in reality, a participant is likely to choose arm B. This same error was also present in the implementation of the Latent-State model.

When correcting the implementation of the τ-Switch and Latent State models, I also checked the implementation of all other models[5]. In doing so, I also noticed an error during my implementation of the likelihood function for the ϵ-decreasing model: where I should have computed the likelihood of a data point when the arm that was chosen is not the "best" arm as $\frac{1-(\epsilon/k)}{N-N_{max}}$, I erroneously calculated it as $\frac{(1-\epsilon)/k}{N-N_{max}}$. This mistake was corrected prior to conducting this year's analysis.

Before conducting this year's analyses, I corrected all three models. I also reran the analysis in Chapter 4 of last year's report; figures which can be compared to last year's results are included in Appendix A. While an in-depth analysis of these results is beyond the scope of this project, a brief comparison of both sets of results shows no major differences in the proportions of models used by participants in the testweek

[4]Thank you to Michael Lee for spotting this error and helping me understand the correct behavior of the model

[5]all models were implemented in Scala

dataset.

The models discussed above were also implemented in JAGS(Plummer et al., 2003) to be used for a fully Bayesian analysis, and so I suspected that I would have to make the same corrections to the JAGS implementations. However, upon inspecting the JAGS code, I found that the models were already implemented according to the correct specification. The Scala and JAGS models were written around three months apart, with the JAGS models being the ones that were implemented last. It may have been the case that with a bit more time with the models lead me to understand them better, and consequently implement them correctly. As such, analyses conducted in last year's project using the JAGS models do not need to be re-tested.

2.4 Relation to Year 2

The result just mentioned, that the majority of participants were inferred to have used the e-WSLS model as their primary decision-making strategy when solving the set of bandit problems, is what has shaped the direction of this project. This result has lead me to develop the following theory: *the e-WSLS model, or a variant thereof, is the dominant decision-making strategy people use when solving bandit problems.* As stated in the introduction, the main aim of this project is to test this theory by attempting to falsify it. In order to do so, I have extended the design optimization framework proposed by Myung and Pitt (2009) in order to produce an optimal experimental design which allowed me to test this theory more rigorously. This framework has two main goals: producing a design where e-WSLS is maximally distinguished from all other models, and that the e-WSLS model should perform sub-optimally compared to those other models under that design. An experiment using this optimal experimental

design was conducted, and the collected data was used to test this theory.

Another interesting piece of work from last year's work was a model and parameter identifiability study conducted on the models I used throughout the project. This model identifiability study allowed me to assess the accuracy of my models, and determine the level of confidence I should have in the results gathered using those models. I believed that a similar study would be of benefit to this year's project also.

Finally, last year I concluded that the extended hierarchical Bayesian mixture model which allowed for per-trial latent switching between models gave the best account of human behavior in bandit problems of all models considered, based on the result on the testweek dataset (Laverty, 2019, p. 103). I also noted in the conclusion of the report that this model was only evaluated on one dataset, and so it would be beneficial to apply it to another dataset to gather more evidence to support this result. This year's work will, therefore, pay close attention to whether the newly developed mixture model also has the largest agreement with human data on the dataset produced by my experiment. Of course, it would be best to design an experiment that would optimally test this model; however, I leave this experiment up to future researchers.

In summary, the main goals of this project were as so:

1. Develop a design optimization framework to produce an optimal experimental design that would allow me to test the theory that the e-WSLS model, or a variant thereof, is the dominant decision-making strategy people use when solving bandit problems.

2. Conduct a model identifiability and parameter recovery study to test the ability of the optimal experimental design to maximally distinguish the e-WSLS from

all others.

3. Conduct an experiment using this optimal experimental design to produce a new dataset of human performance on bandit problems.

4. Apply the models developed last year to this new dataset to test two claims: that the e-WSLS is the dominant decision-making strategy, and that the per-trial extended mixture model gives the best account of human performance on bandit problems.

5. Re-conduct the analysis detailed in Chapter 4 of last year's report with the corrections to the τ-Switch, Latent State, and ϵ-Greedy models.

Chapter 3

Models

As in Chapter 2, I wish to provide readers with the appropriate vocabulary and understanding related to the models used in this project so that they may understand later analyses. Chapter 3 of last year's report includes a detailed explanation of each model, including: the motivation and psychological theory that lead to the development of that model; past studies where it has been applied; a graphical model and a function describing the likelihood of observed data under the model. There are a very limited number of ways in which I can describe a model without repeating the fundamental details necessary to understand it, and there is only one way in which the graphical model and likelihood functions can be described. Any attempt to provide this information again here without repeating myself would not be a productive use of my time, and the end result would likely be a lot less clear than the descriptions I provided last year. As such, as in Chapter 2, I have included, verbatim, the description of each model, along with the graphical models and likelihood functions for each model, from last year's report. All descriptions, figures, and equations in this section should be assumed to have come from last year's report, and credited as so. As in Chapter

2, I have attempted to shorten the descriptions, including only what is necessary for this year's project. For those interested in a discussion of the psychological theory motivating the development of these models, I refer you to Chapter 3 of last year's report (Laverty, 2019). With this point mentioned, and any grounds for plagiarism hopefully dissolved, I will now present the models used in this project.

A note on graphical models and terminology

Graphical models have seen recent use in representing probabilistic generative models in the cognitive sciences (Shiffrin, Lee, Kim, & Wagenmakers, 2008). Zhang and Lee (2010a) report that "[t]he practical advantage of graphical models is that sophisticated and relatively general-purpose Markov Chain Monte Carlo (MCMC) algorithms exist that can sample from the full joint posterior distribution of the parameters conditional on the observed data". Multiple introductions to these models are available (L Griffiths, Kemp, & B Tenenbaum, 2008; M. D. Lee, 2008). In this report, I adopt the formalism presented by M. D. Lee (2008) in which "*... nodes represent variables of interest, and the graph structure is used to indicate dependencies between the variables, with children depending on their parents. The conventions of representing continuous variables with circular nodes and discrete variables with square nodes and of representing unobserved variables without shading and observed variables with shading are used. Stochastic and deterministic unobserved variables are distinguished by using single and double borders, respectively. Plate notation, enclosing with square boundaries subsets of the graph that have independent replications in the model, is also used.*"

Other notation in the graphical models, and likelihood functions, used in this report include:

- D to represent the set of decisions made by a participant, with a decision in-game g and trial k denoted by D_{gk}
- R denoting the rewards received during the task
- S and F denoting the number of successes and failures thus far for a particular arm for the current game

3.1 Guessing

This model is the simplest one considered: given N arms, the model chooses between them uniformly at random. The purpose of this model is to act as a baseline for comparison of other models. The likelihood function for this model is as follows:

$$p(D_k^g = i | M_{guessing}) = \frac{1}{N} \tag{3.1}$$

3.2 Win-Stay Lose-Shift

The Win-Stay Lose-Shift (WSLS) model is a classic model in the reinforcement learning literature (Sutton et al., 1998). In its deterministic form, the model assumes that people stick with a choice while they continue to receive a reward from it, and when the arm no longer pays off, they switch to another. The stochastic version of the model assumes that the participant sticks with a "winning" arm with (high) probability γ, and on any given trial will switch from a winning arm with probability $1 - \gamma$. This parameter, γ, has been described as an "accuracy of execution parameter" (Steyvers et al., 2009), and is useful in describing noisy decision-making data. Many of the models described in this chapter have some form of accuracy of execution parameter

to account for suboptimal decisions in human data. The likelihood function for this model is:

$$p(D_k^g = i|R, \gamma, M_{WSLS}) = \begin{cases} \frac{1}{N} & \text{if } k = 1 \\ \gamma & \text{if } k > 1,\ D_{k-1}^g = i \text{ and } R_{k-1}^g = 1 \\ \frac{1-\gamma}{N-1} & \text{if } k > 1,\ D_{k-1}^g \neq i \text{ and } R_{k-1}^g = 1 \\ 1-\gamma & \text{if } k > 1,\ D_{k-1}^g = i \text{ and } R_{k-1}^g = 0 \\ \frac{\gamma}{N-1} & \text{if } k > 1,\ D_{k-1}^g \neq i \text{ and } R_{k-1}^g = 0 \end{cases} \tag{3.2}$$

Figure 3.1 shows the graphical model for the WSLS model.

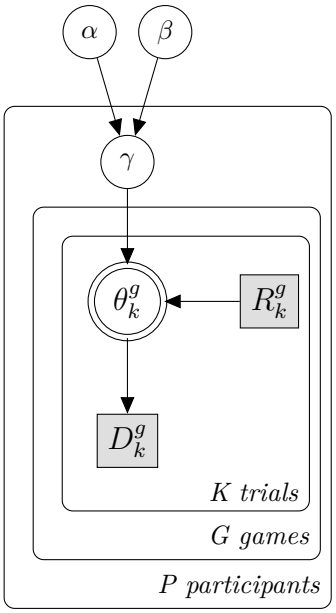


Figure 3.1: Bayesian graphical model for the WSLS model

Figure 3.1 shows how we can allow for individual variation in the "sticking" probability, λ, used by each participant, while assuming commonality in all participants by assuming this parameter comes from some underlying process, described here by a

Beta distribution, parameterized by α and β. The use of a Beta distribution for modeling human assumptions about the environment is standard practice when working with bandit problems (Steyvers et al., 2009). One reason for using a Beta distribution is that it allows for an intuitive explanation of how we believe people might think about the environment. In essence, the two parameters used, α and β, can be thought of as a count of prior successes and prior failures, respectively. With these parameters at hand, we can construct psychological assumptions about the level of optimism a player has about the environment as $\frac{\alpha}{\alpha+\beta}$, and the level of certainty they have in their optimism as $\alpha+\beta$. We should expect that people will behave very differently based on these prior assumptions on the environments: where levels of optimism are high, we might expect higher levels of exploration as participants believe there are many high paying arms. Conversely, in situations were optimism is lower, or participants are less certain in their assumptions, they may explore less often and choose to stick with whichever arm they have relatively more information on. Once a model has been fit to the data, we can analyze the posterior estimates of α and β to make inferences about participant's assumptions about the environment, as will be seen in later analyses.

3.3 Extended Win-Stay Lose-Shift

The extended Win-Stay Lose-Shift (e-WSLS) is very similar to the original WSLS model described above, except that the probability of staying with a winning arm and the probability of switching from a losing arm are no longer parameterized by the same probability γ: the probability of staying with a winning arm is γ_w, while the probability of switching from a losing arm is now γ_l. The likelihood function, and graphical model, for this model are shown below:

$$p(D_k^g = i|R, \gamma_w, \gamma_l, M_{e-WSLS}) = \begin{cases} \frac{1}{N} & \text{if } k = 1 \\ \gamma_w & \text{if } k > 1,\ D_{k-1}^g = i \text{ and } R_{k-1}^g = 1 \\ \frac{1-\gamma_w}{N-1} & \text{if } k > 1,\ D_{k-1}^g \neq i \text{ and } R_{k-1}^g = 1 \\ 1 - \gamma_l & \text{if } k > 1,\ D_{k-1}^g = i \text{ and } R_{k-1}^g = 0 \\ \frac{\gamma_l}{N-1} & \text{if } k > 1,\ D_{k-1}^g \neq i \text{ and } R_{k-1}^g = 0 \end{cases} \tag{3.3}$$

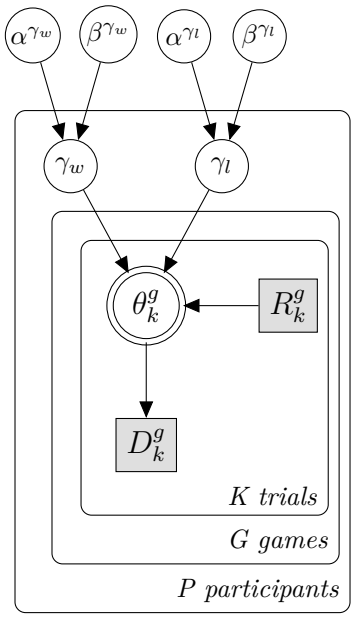


Figure 3.2: Bayesian graphical model for e-WSLS model

3.4 ϵ-Greedy

On any given trial, the ϵ-Greedy model[1] chooses an arm at random according to probability ϵ, otherwise it chooses the "best" arm. In order to choose the "best"

[1]In the paper by Steyvers et al. (2009), the ϵ-greedy model is used under the name Success Ratio. Where results from last year are presented, any reference to the Success Ratio model should be understood as describing the ϵ-Greedy model

arm in this context, the model maintains a record of the proportion of successes and failures of each arm, and chooses an arm based on it's ratio of successes to failures. The likelihood function for the ϵ-greedy model is as follows:

$$p(D_k^g = i|S, F, \epsilon, M_{\epsilon-greedy}) = \begin{cases} \frac{1}{N} & \text{if } k = 1 \\ \frac{\epsilon}{N_{max}} & \text{if } k > 1 \text{ and } i \in \arg\max_j \frac{S_j+1}{S_j+F_j+2} \\ \frac{1-\epsilon}{N-N_{max}} & \text{otherwise} \end{cases} \tag{3.4}$$

Figure 3.3 shows the graphical model for the ϵ-Greedy model.

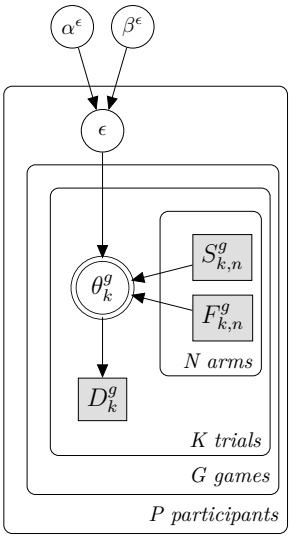


Figure 3.3: Bayesian graphical model for the ϵ-Greedy model

3.5 ϵ-Decreasing

The ϵ-decreasing model is a simple variant of the ϵ-greedy model presented above. The key difference between the two models is that the value of ϵ remains fixed in the ϵ-greedy model, whilst it decreases as time progresses in the ϵ-decreasing variant. In this project, ϵ is decreased linearly with time. The likelihood function for this model is:

$$p(D_k^g = i|S, F, \epsilon, M_{\epsilon-decreasing}) = \begin{cases} \frac{1}{N} & \text{if } k = 1 \\ \frac{\epsilon/k}{N_{max}} & \text{if } k > 1 \text{ and } i \in \arg\max_j \frac{S_j+1}{S_j+F_j+1} \\ \frac{1-(\epsilon/k)}{N-N_{max}} & \text{otherwise} \end{cases} \quad (3.5)$$

and the graphical model is shown in Figure 3.4:

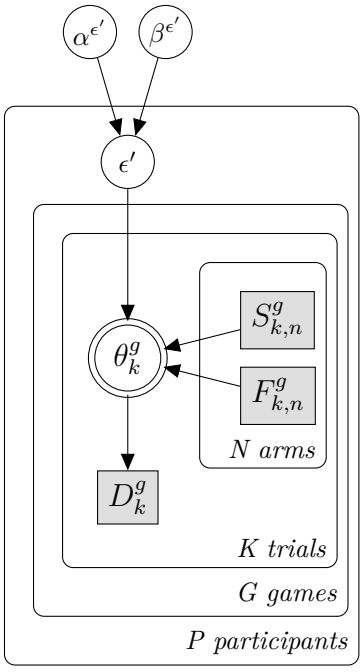


Figure 3.4: Bayesian graphical model for the ϵ-Decreasing model

3.6 π-First

The π-First model describes decision-making as taking place in two distinct phases: an exploratory phase in which all decisions are made at random, and exploitative phase where the best arm is greedily chosen. The time spent in the latent "explore" phase is controlled by the parameter π, which dictates after which trial number the participant switches to the "exploit" phase. In the stochastic version of the model, an accuracy of execution parameter, γ, is included. The likelihood function for this model is:

$$p(D_k^g = i|S, F, \pi, \gamma, M_{\pi-first}) = \begin{cases} \frac{1}{N} & \text{if } k \leq \pi \\ \frac{\gamma}{N_{max}} & \text{if } k > \pi \text{ and } i \in \arg\max_j \frac{S_j+1}{S_j+F_j+1} \\ \frac{1-\gamma}{N-N_{max}} & \text{otherwise} \end{cases} \quad (3.6)$$

Figure 3.5 shows the graphical model for the π-first model:

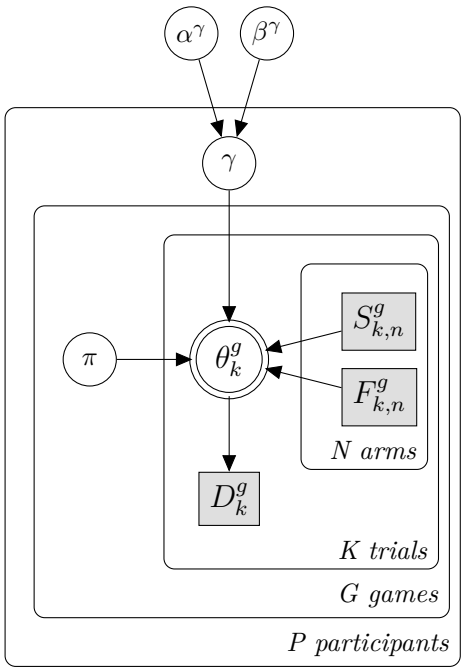


Figure 3.5: Bayesian graphical model for for the π-first model

3.7 Latent State and τ-Switch Models

3.7.1 Latent State Model

The Latent State Model extends the π-First model to allow for repeated switching between the two latent states, rather than switching once from exploration to exploitation as outlined in the π-first model. The other difference between the π-first and the latent state model is the mechanism by which the model makes "explore" and "exploit" decisions. In the π-first model, exploring consists of choosing an arm at random, while exploiting means greedily choosing the arm with the highest proportion of successes encountered so far. The new latent state model proposed distinguishes between 3 different situations, and acts according to both the latent state - explore or exploit - and the situation the model finds itself in. The three situations are described in a paper by the original author of the model as follows:

"In the same situation, both alternatives have the same number of observed successes and failures. ... For the same situation, both alternatives have an equal probability of being chosen."

"In the better–worse situation, one alternative has more successes and fewer failures than the other alternative (or more successes and equal failures, or equal successes and fewer failures). In this situation, one alternative is clearly better than the other. ... For the better–worse situation, the better alternative has a high probability, given by a parameter γ, of being chosen. The probability the worse alternative is chosen is $1 - \gamma$."

"In the explore–exploit situation, one alternative has been chosen more often, and has more successes but also more failures than the other alternative. In this situation,

neither alternative is clearly better, and the decision-maker faces the explore–exploit dilemma. Choosing the better-understood alternative corresponds to exploiting, while choosing the less well-understood alternative corresponds to exploring. ... In this situation, our model assumes the exploration alternative will be chosen with the high probability γ if the decision-maker is in a latent 'explore' state, but the exploitation alternative will be chosen with probability γ if the decision-maker is in the latent exploit state. In this way, the latent state for a trial controls how the exploration versus exploitation dilemma is solved at that stage of the problem." (M. D. Lee et al., 2011)

3.7.2 τ-Switch

In a follow-up paper, M. D. Lee et al. (2009) simplified the model to be more similar to the original π-first model, removing the latent state parameter for each trial and instead incorporating a single switch-point parameter, τ, giving rise to the τ-switch model. This model is now more similar to the π-first model in that a single parameter encodes the switching point, however, it is still fundamentally different from the π-first model due to it's modeling of three different situations which are used to control the model's decision making.

3.7.3 Extending the models to N-armed bandit problems

In order to define the Latent State and τ-switch model for N arms, the following formalism was developed as part of last year's project:

To begin with, it is helpful to define two sets: S and F, where S consists of all of the

arms who have a number of successes equal to the maximum number of successes of all arms s, and F consists of all of the arms with a number of failures equal to the minimum number of failures across all arms, f. Each of these sets is calculated at the beginning of each trial based on previous successes and failures. These sets can then be used to determine the behavior of the model in each of the three situations outlined in the original paper.

Same situation

Rather than considering only whether two arms have the same number of successes and failures, the model is in the same situation if two *or more* arms appear in both S and F, i.e., two or more arms have both the maximum number of successes, and the minimum number of failures, therefore all alternatives have an equal probability of being chosen.

Better-Worse situation

We encounter the better-worse situation if only one arm is in both S and F, meaning only one arm has had both more successes **and** fewer failures than all other arms, so it is clearly the best choice. This arm is therefore chosen with probability γ, or one of the worse alternatives are chosen with probability $\frac{1-\gamma}{N-1}$.

Explore-Exploit situation

The explore-exploit situation arises when $S \cap F = \emptyset$, and both S and F contain at least one element. This means that at least one arm has had more successes than all other arms, but also more failures, meaning more information is available for these arms and so choosing one of these arms would constitute and exploit decision. In line

with the correction to the models discussed in Chapter 2, if more than one arm is in S, the model with the minimum number of failures among these arms is preferred. If there are multiple arms in S with the same minimum number of failures, they each have the same probability of being chosen. To present these additional criteria in the likelihood function of the τ-Switch and Latent State models, I will define the subset of arms in S with the minimum number of failures across the arms in S, f_S - **not** to be confused the with the overall minimum number of failures across *all* arms, f - as S_{f_S}. Alternatively, choosing an arm from those in F - the ones we have less information on - would constitute an explore decision. Like in the exploit cases, we can add additional choosing criteria to this explore case: if there is more than one arm in F, the arm with the maximum number of successes would likely be chosen. If more than one such arm exists, each has an equal probability of being chosen. I will define F_{s_F} as the subset of arms in F with number of success equal to the maximum number of the success across all arms in F, s_F - again, **not** to be confused with the maximum number of successes across *all* arms, s.

Again, an accuracy of execution parameter, γ is included in the model to allow for sub-optimal decisions. Note that this situation is different from the better-worse situation since the arm with the greatest number of success no longer also has the minimum number of failures.

The likelihood function for the τ-switch model for N-armed bandits is presented as follows, with the latent state model following a similar structure with τ replaced with a latent switching parameter z:

$$p(D_k^g = i|S, F, \tau, \gamma, M_{\tau-switch}) = \begin{cases} \frac{1}{|S_k \cap F_k|} & \text{if } i \in S_k \cap F_k, \text{ and } |S_k \cap F_k| > 1 \\ \gamma & \text{if } i \in S_k \cap F_k, \text{ and } |S_k \cap F_k| = 1 \\ \frac{1-\gamma}{N-1} & \text{if } i \notin S_k \cap F_k, \text{ and } |S_k \cap F_k| = 1 \\ \frac{\gamma}{|F_{s_F,k}|} & \text{if } k \leq \tau,\ i \in F_{s_F,k}, \text{ and } |S_k \cap F_k| = 0 \\ \frac{1-\gamma}{|F_{s_F,k}|} & \text{if } k > \tau,\ i \in F_{s_F,k}, \text{ and } |S_k \cap F_k| = 0 \\ \frac{1-\gamma}{|S_{f_S,k}|} & \text{if } k \leq \tau,\ i \in S_{f_S,k}, \text{ and } |S_k \cap F_k| = 0 \\ \frac{\gamma}{|S_{f_S,k}|} & \text{if } k > \tau,\ i \in S_{f_S,k}, \text{ and } |S_k \cap F_k| = 0 \end{cases} \tag{3.7}$$

Figure 3.6 shows the graphical model for the τ-switch model.

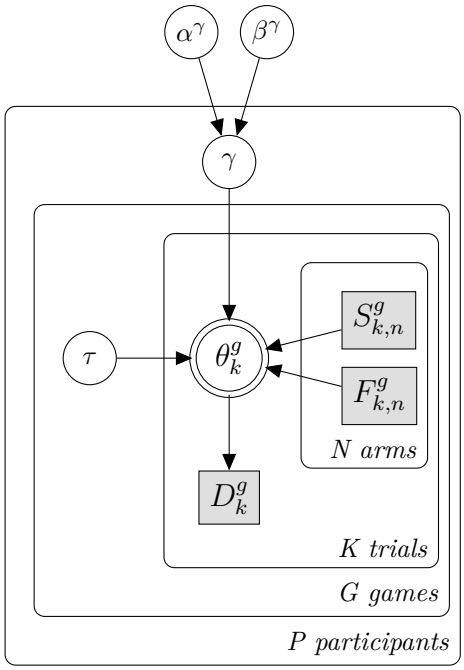


Figure 3.6: Bayesian graphical model for the τ-switch model

3.8 Optimal

For a fixed environment, we can calculate the optimal decision process according to the dynamic programming method outlined by Kaelbling, Littman, and Moore (1996). The expense of calculating the optimal decision policy for a bandit problem is exponential, however, for a small enough number of trials and arms, the computation is feasible.

The idea behind the method, as is the case with all dynamic programming methods, is that we choose a base case, in this case the last trial, and assume that on the final trial the arm with the greatest expected reward is chosen. From here, we choose the arm on the second-last trial which gives the greatest expected reward over the final two trials, assuming that the final decision is made optimally. We iterate backward through the sequence of trials in this manner, and in the end, we will have an optimal sequence of decisions for the entire problem.

In order to calculate the optimal policy, we must compute a mapping between belief states and actions (Kaelbling et al., 1996). A belief state is a representation of the information available to the agent: Kaelbling et al. (1996) define this as a vector containing the number of times an arm has been pulled, n and the rewards, w received so that a belief state takes the form $\{n_1, w_1, ..., n_k, w_k\}$; Steyvers et al. (2009) use a slightly different notation where a belief state for a given game, g, and given trial, k, is represented as the number of successes and failures for each arm, denoted s_i and f_i respectively for arm i so that the belief state is of the form: $\{s_1, f_1, ..., s_N, f_N\}$. I will adopt the latter notation for the remainder of this report for consistency, as the work of Steyvers et al. (2009) is the subject of the replication study described later.

The expected additional reward to be gained by acting optimally for the remainder of trials, given that we are currently at trial k, is denoted $V_k^*(s_1, f_1, ..., s_N, f_N)$, where the expected additional reward for the final trial $V_K^*(s_1, f_1, ..., s_N, f_N) = 0$. The value of V_k^* for any other trial can be calculated recursively according to the following equation, with the probability of success and failure on each trial defined for Beta(α, β) environments as defined by Steyvers et al. (2009):

$$V_k^*(s_1, f_1, ..., s_N, f_N) = \max_i E\left[\begin{array}{l}\text{Future reward if the agent chooses arm i} \\ \\ \text{and then acts optimally for the remainder of pulls}\end{array}\right]$$

$$= \max_i \left[\begin{array}{l}\frac{s_i + \alpha}{s_i + f_i + \alpha + \beta} V_{k+1}^*(s_1, f_1, ..., s_{i+1}, f_i, s_N, f_N) + \\ \frac{f_i + \beta}{s_i + f_i + \alpha + \beta} V_{k+1}^*(s_1, f_1, ..., s_i, f_{i+1}, s_N, f_N)\end{array}\right]$$

Steyvers et al. (2009) present the likelihood for data under the optimal model as follows, where α and β the participants assumptions about the environment, and w acts as an accuracy of execution parameter:

$$p(D_k^g = i | S, F, \alpha, \beta, w, M_{opt}) = \begin{cases} \frac{w}{N_{max}} & \text{if the } i^{th} \text{ alternative maximises total expected reward} \\ \\ \frac{1-w}{N-N_{max}} & \text{otherwise} \end{cases} \tag{3.8}$$

3.9 Hierarchical Mixture Model

Zhang and Lee (2010a) develop a hierarchical mixture of four decision-making models - WSLS, e-WSLS, ϵ-greedy, and ϵ-decreasing - in order to accommodate for differences not only the parameters used for a given model, but for differences in the model

used by each participant. The mixture parameter, z_p, indicates which model the p^{th} participant is thought to be using. The assignment parameter, z_p, has prior $z_p \sim Categorical(\phi)$ with a uniform Dirichlet prior placed on ϕ so no model one is preferred. In an analysis of model differences, the posterior expectation of ϕ is analyzed to infer the proportion of participants using each model.

Figure 3.7 shows the graphical model for this mixture model (with Beta, rather than Gaussian priors).

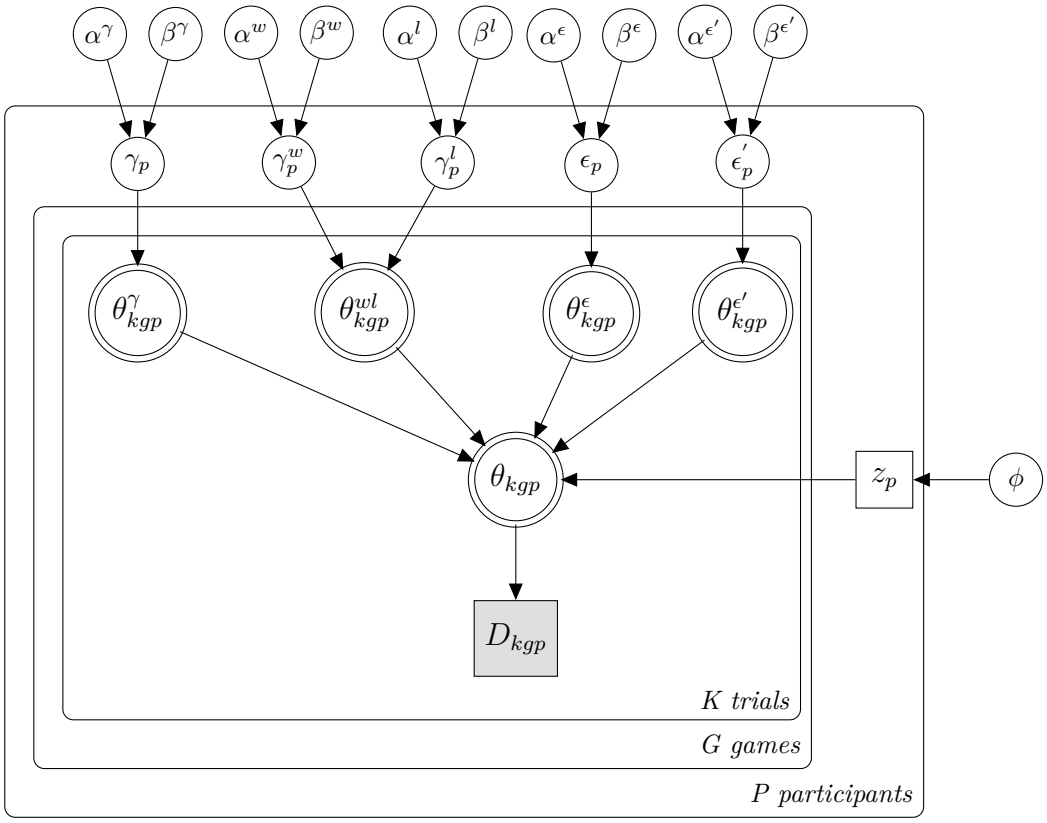


Figure 3.7: Bayesian graphical model for mixture model proposed in (Zhang & Lee, 2010a)

3.10 Extended Hierarchical Mixture Model

In last year's project, I extended the mixture model presented above to include a wider range of component models: the τ-Switch, Latent State, and π-First models were added.

Figure 3.8 shows the graphical model for the proposed mixture model, with latent switch parameters for the τ-switch and π-first models excluded for the sake of readability.

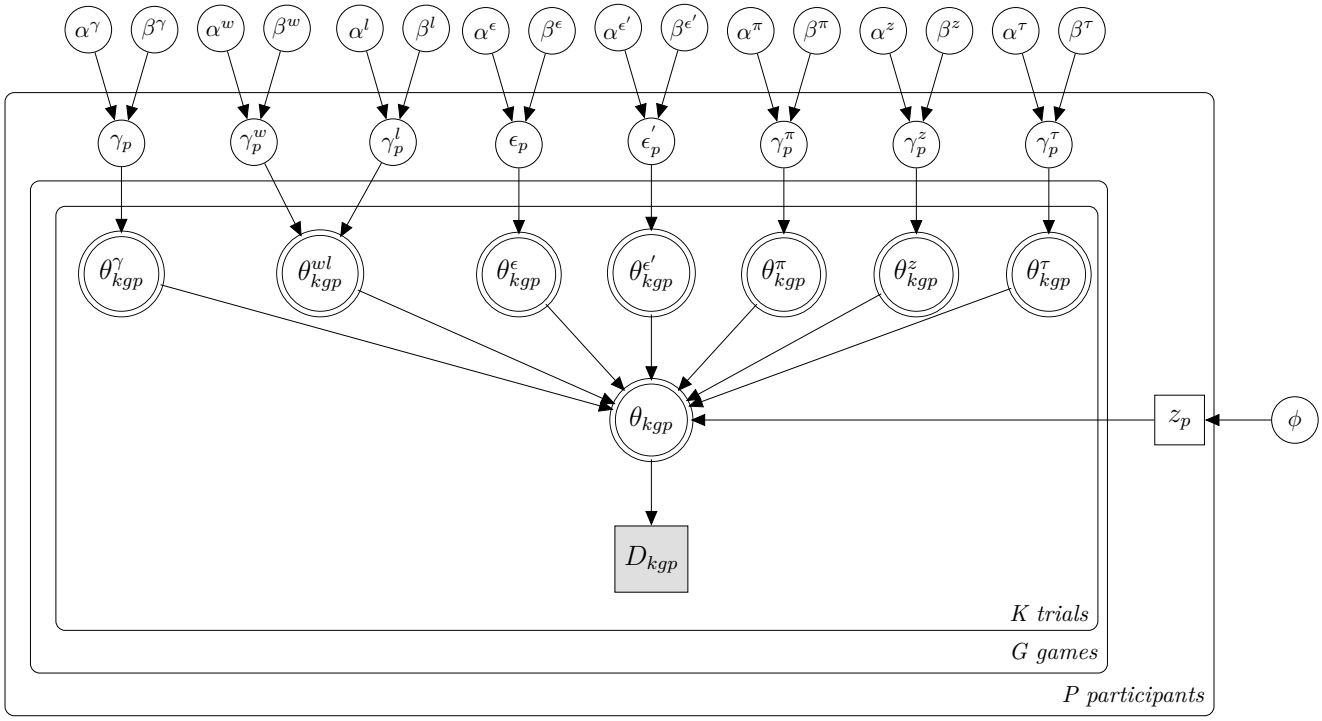


Figure 3.8: Bayesian graphical model for the newly proposed Extended Mixture Model

3.11 Extensions to Mixture Models

In last year's project, I also proposed what I believed to be two natural extensions to the mixture models presented in sections 3.10 and 3.11. These extensions included allowing participants to switch to a different decision-making policy on either each

new game or each new trial encountered.

For the sake of brevity, I have not included graphical models for the newly proposed models as they are very similar to the graphical models presented in Figures 3.7 and 3.8. The only difference in the new models is that the latent allocation parameter, z is moved inside of the *game* plate and *trial* plate for the per-game and per-trial extensions, respectively.

A note on naming conventions for the mixture models

Already in this report I have made frequent reference to the mixture models described in sections 3.9 - 3.11, and I will continue to do so throughout this project. I'm sure you can agree at this point that referring to a model as the *"extended hierarchical Bayesian mixture model which allows for latent switching between models on a per-trial basis"* is a bit of a mouthful. As such, I will use the following shortened names for the models: the mixture model proposed by (Zhang & Lee, 2010a) will be referred to as the *MM*; the extension to this model described in section 3.10 will be referred to as the *e-MM*. The variants which allow for latent switching on per-trial and per-game bases will be referred to as the *(per-trial/per-game) (MM/e-MM)*. I hope that this naming convention clarifies, rather than further obscures, this report.

Chapter 4

Optimal Experimental Design

4.1 Designing a utility function

In their original paper, Myung and Pitt (2009) propose a framework for design optimization which involves finding the design $\boldsymbol{d}^*$ that maximises some utility function $U(\boldsymbol{d})$ which quantifies the quality of a design $\boldsymbol{d}$; what exactly constitutes this design will be discussed shortly. The authors originally introduced the design optimization (DO) framework for the task of finding designs which have the greatest likelihood of differentiating between models under consideration, particularly for mathematical models used in psychology. The framework was applied to the task of finding optimal experiments which distinguished between models of retention, and models of categorization (Myung & Pitt, 2009). Zhang and Lee (2010b) later applied this framework to models of human decision making on bandit problems; finding bandit designs which maximally differentiated between different psychological models of human decision making.

The original paper by Myung and Pitt proposed a utility function based upon fitting

one model on data, $\boldsymbol{y}_A$, generated by another model, according to the latter model's parameters, $\boldsymbol{\theta}_A$. The fitting procedure they proposed was minimizing the sum of square errors:

$$u(\boldsymbol{d}, \boldsymbol{\theta}_A, \boldsymbol{y}_A) = \sum_{i=1}^{N} (y_{Ai} - prd_{Bi}(\boldsymbol{\theta}_B^*, \boldsymbol{d}))^2 \tag{4.1}$$

where $prd_{Bi}(\boldsymbol{\theta}_B^*, \boldsymbol{d})$ is the prediction by model B at its best-fitting parameter vector $\boldsymbol{\theta}_B^*$ given the design $\boldsymbol{d}$. Of course, when designing an experiment we do not yet have the observed data, $\boldsymbol{y}_A$, nor the parameter values, $\boldsymbol{\theta}_A$, responsible for generating this data. The quality of the design, $\boldsymbol{d}$, is therefore calculated by the expectation of $u(\boldsymbol{d}, \boldsymbol{\theta}_A, \boldsymbol{y}_A)$ with respect to the sampling distribution $p(\boldsymbol{y}_A|\boldsymbol{\theta}_A, \boldsymbol{d})$ and the prior distribution $p(\boldsymbol{\theta}_A|\boldsymbol{d})$, for a single model M_A:

$$U(\boldsymbol{d}) = \iint u(\boldsymbol{d}, \boldsymbol{\theta}_A, \boldsymbol{y}_A) p(\boldsymbol{y}_A|\boldsymbol{\theta}_A, \boldsymbol{d}) p(\boldsymbol{\theta}_A|\boldsymbol{d})\, d\boldsymbol{\theta}_A\, d\boldsymbol{y}_A \tag{4.2}$$

The model above, therefore, is an expression of the expected "badness-of-fit" of model B conditional on data generated from model A. Myung and Pitt (2009) go on to present a full expression for the utility of a design as one which also accounts for the "badness-of-fit" of model A on data generated by model B, since we don't know which of the two models will generate an experimental outcome:

$$\begin{aligned} U(\boldsymbol{d}) = p(M_A) \iint u(\boldsymbol{d}, \boldsymbol{\theta}_A, \boldsymbol{y}_A) p(\boldsymbol{y}_A|\boldsymbol{\theta}_A, \boldsymbol{d}) p(\boldsymbol{\theta}_A|\boldsymbol{d})\, d\boldsymbol{\theta}_A\, d\boldsymbol{y}_A \\ + p(M_B) \iint u(\boldsymbol{d}, \boldsymbol{\theta}_B, \boldsymbol{y}_B) p(\boldsymbol{y}_B|\boldsymbol{\theta}_B, \boldsymbol{d}) p(\boldsymbol{\theta}_B|\boldsymbol{d})\, d\boldsymbol{\theta}_B\, d\boldsymbol{y}_B \end{aligned} \tag{4.3}$$

where $u(\boldsymbol{d}, \boldsymbol{\theta}_B, \boldsymbol{y}_B)$ is adjusted so that model A is fit to data generated by model B, and where where $p(M_A)$ and $p(M_B)$ are the prior probabilities of the two models. By maximizing $U(\boldsymbol{d})$, the design which gives the maximum "badness-of-fit" of each model on data generated by the other will be chosen, thereby producing a design which maximally differentiates between the two models. To disambiguate between

the two utility functions $u(.)$ and $U(.)$, the authors dub the former the *local utility function*, and the latter the *global utility function*; I will adopt this terminology here too.

Zhang and Lee (2010b) adapt this framework to be used on bandit tasks. They point out that the local utility function should reflect the likelihood of the "correct" model based on observed data, and thus use the Bayes Factor as their local utility function, as it is a standard in Bayesian model selection (Kass & Raftery, 1995). The Bayes factor is the ratio of the marginal likelihood of two competing hypotheses; here the hypotheses are that the observed data was generated by each model. By computing the Bayes factors for models A and B on data generated by model A as so:

$$BF_{A/B} = \frac{\int p(\boldsymbol{y}_A|\boldsymbol{\theta}_A)p(\boldsymbol{\theta}_A)\,d\boldsymbol{\theta}_A}{\int p(\boldsymbol{y}_a|\boldsymbol{\theta}_B)p(\boldsymbol{\theta}_B)\,d\boldsymbol{\theta}_B} \tag{4.4}$$

Using Bayes factors as our local utility function, maximizing the global utility function will produce a design, $\boldsymbol{d}$, in which model A gave the worst fit to data generated by model B, and vice versa for model B, thus producing a design which maximally differentiates between models.

So far, the discussion has only been concerned with utility functions which maximally differentiate between 2 models, however, as we have seen in Chapter 3, there are many models of human decision making on bandit problems. How then can we extend the framework presented by Myung and Pitt (2009) to account for such a use case? The original authors note that "it is straightforward to modify [Equation 4.3] to accommodate the situation in which more than two models are to be discriminated", however, they stick to the case of 2 models in all of their case studies. Zhang and Lee (2010b) offer a more general form of Equation 4.2, marginalising over the potentially

infinite model space, $\mathcal{M}$:

$$U(\boldsymbol{d}) = \iiint u(\boldsymbol{d}, \boldsymbol{\theta}, \boldsymbol{y}) p(\boldsymbol{y}|\boldsymbol{\theta}, \boldsymbol{d}) p(\boldsymbol{\theta}|\boldsymbol{d}) \, d\boldsymbol{\theta} \, d\boldsymbol{y} \, d\mathcal{M} \tag{4.5}$$

While they present this general framework, they too consider only the case of comparing two models at a time - while they do examine three models in total, only two models were ever compared at a time. I will, therefore, attempt to provide an concrete implementation of the DO framework which maximally differentiates between multiple models. As a first port of call, I will place a limit on the potentially infinite model space, considering instead a finite set of models, M, in which $M \subset \mathcal{M}$. As such, we can rewrite Equation 4.5 as a sum over the finite subset of models, M, and calculate $U(\boldsymbol{d})$ like so:

$$U(\boldsymbol{d}) = \sum_{m \in M} p(m) \iint u(\boldsymbol{d}, \boldsymbol{\theta}_m, \boldsymbol{y}_m) p(\boldsymbol{y}_m|\boldsymbol{\theta}_m, \boldsymbol{d}) p(\boldsymbol{\theta}_m|\boldsymbol{d}) \, d\boldsymbol{\theta}_m \, d\boldsymbol{y}_m \tag{4.6}$$

The question now arises as to how we should design the local utility function, $u(\boldsymbol{d}, \boldsymbol{\theta}_m, \boldsymbol{y}_m)$, since all those considered thus far have dealt with only 2 models. I propose the following utility function[1]:

$$u(\boldsymbol{d}, \boldsymbol{\theta}_m, \boldsymbol{y}_m) = \sum_{m' \in M, m' \neq m} BF_{m/m'} \tag{4.7}$$

In this framework we'll generate data according to some reference model, and compute the Bayes factor of that model against all other models, summing the individual Bayes factors to give a final value for the utility of the design relative to that reference model; maximising this value will give a design in which the "correct" model is the most likely out of all other models.[2] We will then repeat this data generating and

[1]The case where $m' = m$ is ignored since this will always result in a Bayes factor of 1

[2] A word of warning to go along with this local utility function: I believe it is possible that the

evaluation procedure for all models in M, resulting in the global utility function:

$$U(\boldsymbol{d}) = \sum_{m \in M} p(m) \iint \sum_{m' \in M, m' \neq m} BF_{m/m'} p(\boldsymbol{y}_m | \boldsymbol{\theta}_m, \boldsymbol{d}) p(\boldsymbol{\theta}_m | \boldsymbol{d}) \, d\boldsymbol{\theta}_m \, d\boldsymbol{y}_m \tag{4.8}$$

By taking $\boldsymbol{d}^* = \arg\max_{\boldsymbol{d} \in \mathcal{D}} U(\boldsymbol{d})$, the design produced will maximally differentiate between all models in M. The purpose of this project is to determine whether people predominantly use the e-WSLS strategy in when solving bandit problems, therefore, I wish to pay special attention to designs concerned with differentiating the e-WSLS model from all others. Solving Equation 4.8 will produce a design in which *all* models are maximally differentiated, to see how this may be a problem for my project I present a (simplified) hypothetical example:

Consider two designs, $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$, and three models, e-WSLS, π-First, and ϵ-greedy. Lets say the local utility function for one set of data generated under the e-WSLS model is calculated with the following Bayes factors: $\text{BF}_{\frac{e-WSLS}{\epsilon-Greedy}} = 5.0$, $\text{BF}_{\frac{e-WSLS}{\pi-First}} = 5.0$, so $u(\boldsymbol{d}_1, \boldsymbol{\theta}_{e-WSLS}, \boldsymbol{y}_{e-WSLS}) = 10.0$. Let's do the same for data generated under the ϵ-Greedy model: $\text{BF}_{\frac{\epsilon-Greedy}{e-WSLS}} = 5.0$, $\text{BF}_{\frac{\epsilon-Greedy}{\pi-First}} = 5.0$, so $u(\boldsymbol{d}_1, \boldsymbol{\theta}_{\epsilon-Greedy}, \boldsymbol{y}_{\epsilon-Greedy}) = 10.0$. Finally, for data generated by the π-First model: $\text{BF}_{\frac{\pi-First}{e-WSLS}} = 5.0$, $\text{BF}_{\frac{\pi-First}{\epsilon-Greedy}} = 5.0$, so $u(\boldsymbol{d}_1, \boldsymbol{\theta}_{\pi-First}, \boldsymbol{y}_{\pi-First}) = 10.0$. The overall global utility $U(\boldsymbol{d}_1) = 30.0$ (assuming for the sake of simplicity in this example that only one set of parameters, and one set of data per model are generated for a given design).

Imagine we then calculate the global utility of $\boldsymbol{d}_2$ and get the following results: $\text{BF}_{\frac{e-WSLS}{\epsilon-Greedy}} = 0.5$, $\text{BF}_{\frac{e-WSLS}{\pi-First}} = 0.5$, $\text{BF}_{\frac{\epsilon-Greedy}{e-WSLS}} = 0.5$, $\text{BF}_{\frac{\epsilon-Greedy}{\pi-First}} = 15.0$, $\text{BF}_{\frac{\pi-First}{e-WSLS}}$

value of the local utility function could be skewed by extreme Bayes factors for one model dominating all others. While this is possible, empirical results from experiments discussed in section 4.3.2 show that this was not the case for this project, however, researchers must be aware of the possibility of extreme Bayes factor values, and resulting extreme local utility function values, when searching for optimal designs.

= 0.5, $\text{BF}_{\frac{\pi-First}{\epsilon-Greedy}} = 15.0$. The final value for $U(\boldsymbol{d}_2) = 32.0$, and so $\boldsymbol{d}_2$ is chosen as the optimal experimental design. This example draws attention to a potential danger that may arise when relying solely final utility function values - as discussed briefly in footnote 2 - that some extreme Bayes factor results may skew the results in favour of outlying models; in this case the π-First and ϵ-Greedy models. As mentioned already, empirical results suggest that such extreme cases are very unlikely, however, experimenters must take caution when analysing the results of such experimental design frameworks, and pay attention not only to the final global utility value, but also the component utility functions.

Additionally, and the main takeaway from this example, is that even in the face of not wildly extreme values, some designs may differentiate better between a subset of models than another. In this case, $\boldsymbol{d}_2$ presents an experiment where it is easy to distinguish between ϵ-Greedy and π-First models, but not so much between any results involving the e-WSLS model. If we want to test the e-WSLS, we'd want to choose $\mathbf{d}_1$. As mentioned before I presented this example, the purpose of this project is to determine whether people predominantly use the e-WSLS strategy in when solving bandit problems and so I wish to find designs which maximally differentiate the e-WSLS model from all others. I, therefore, present a modified version of Equation 4.8 which finds a design which maximally differentiates between a chosen model m_c, and all other models in M:

$$\begin{aligned} U(\boldsymbol{d}; m_c) =& p(m_c) \iint \sum_{m' \in M, m' \neq m_c} BF_{m_c/m'} p(\boldsymbol{y}_{m_c} | \boldsymbol{\theta}_{m_c}, \boldsymbol{d}) p(\boldsymbol{\theta}_{m_c} | \boldsymbol{d}) \, d\boldsymbol{\theta}_{m_c} \, d\boldsymbol{y}_{m_c} \\ &+ \sum_{m \in M, m \neq m_c} p(m) \iint BF_{m/m_c} p(\boldsymbol{y}_m | \boldsymbol{\theta}_m, \boldsymbol{d}) p(\boldsymbol{\theta}_m | \boldsymbol{d}) \, d\boldsymbol{\theta}_m \, d\boldsymbol{y}_m \end{aligned} \tag{4.9}$$

We can see in the equation above that all BF calculations that do not involve the model of choice m_c are omitted. While doing so sacrifices some ability to differentiate

all models from one another, we gain the assurance that the design produced will clearly distinguish the model we are most concerned with from all others. Equation 4.9 has the additional side-effect of reducing the number of calculations necessary to compute the value of the global utility function, something which will be of enormous benefit considering the large space of design and model parameters we have to consider. To my knowledge, this is the first application of the framework presented by Myung and Pitt (2009) to give preference to one model over all others.

4.2 Computational method

While we now have a description of the function to be optimized in order to produce an optimal design, $\boldsymbol{d}^*$, it is clear maximizing $U(\boldsymbol{d}; m_c)$ is a nontrivial problem, as Equation 4.9 involves the evaluation of a double integral. It is, therefore, not generally possible to find an analytic solution to Equation 4.9, and so we must turn to computational methods. Multiple methods have been proposed to find a solution to the design optimization problem, with two plausible paths being: a grid search over the data and parameter spaces, and simulation-based approaches developed in statistics (Myung & Pitt, 2009; Zhang & Lee, 2010b). While the details of the simulation-based approaches are interesting, they are not relevant to this project; interested readers should see the papers mentioned above for such details. Of relevance here is the grid search approach, and so the details of how that method was adapted for this project will now be provided.

4.2.1 Grid method

One approach for a solution to the design optimization problem is to approximate the integrals in Equation 4.9 using a grid search. The advantage of using a grid search instead of statistical approaches is that it is more straightforward to understand, as well as to implement. In addition to this, a grid search will consider the entire space of possible designs and is immune to the problem of local optima. The downside, of course, is that it is very computationally expensive; this is the main reason previous researchers have opted for simulation-based approaches (Zhang & Lee, 2010b). In this section, I will present details of the grid approximation for the problem of design optimization described in Equation 4.9. In the next section I will present the technical details of the implementation of the grid search, as well as the techniques I used to overcome the computational inefficiency of the method so that it could be used for this project.

Zhang and Lee (2010b) are the first to seriously consider the use of grid approximation to solve the optimal design problem for bandit problems. In their consideration, they present pseudo-code for how the method could be applied to the problem at hand; the pseudo-code they presented specifically addressed the problem of producing an optimal design to distinguish between the e-WSLS and ϵ-Greedy models. I have adapted their pseudo-code so that it, instead, describes the process of finding an optimal design which distinguishes a *single, chosen*, model from a set of M candidate models as discussed in Equation 4.9. The pseudo-code for this approach is presented in Algorithm 1.

The pseudo-code describes the process for finding the optimal *design* which *maxi-*

Algorithm 1 Pseudo-code for the grid approach design optimization for bandit problems, for the problem of finding the optimal design to distinguish a chosen model, m_c, from a set of candidate models, M

Choose a set of candidate designs, D

for all designs $\mathbf{d} \in D$ **do**

for $i = 1$ to J samples **do**

sample parameters for m_c, $\boldsymbol{\theta}_{m_c} \sim \text{Beta}(\boldsymbol{\alpha}_{m_c}, \boldsymbol{\beta}_{m_c})$

sample data for current experimental design using m_c, $\mathbf{y}_{m_c} \sim p_{m_c}(\boldsymbol{\theta}_{m_c}, \boldsymbol{d})$

calculate $u(\boldsymbol{d}, \boldsymbol{\theta}_{m_c}, \boldsymbol{y}_{m_c})$ for data $\mathbf{y}_{m_c}$

for each model m in the list of all models of interest M, $m \neq m_c$ **do**

sample parameter(s) for model m, $\boldsymbol{\theta}_m \sim \text{Beta}(\boldsymbol{\alpha}_m, \boldsymbol{\beta}_m)$

sample data for current experimental design using model m, $\mathbf{y}_m \sim p_m(\boldsymbol{\theta}_m)$

calculate BF_{m/m_c} for data $\mathbf{y}_m$

end for

calculate $U_i(\mathbf{d}) = u(\boldsymbol{d}, \boldsymbol{\theta}_{m_c}, \boldsymbol{y}_{m_c}) + \sum_{m \in M} BF_{m/m_c}$

end for

estimate utility of design as $\hat{U}(\mathbf{d}) = 1/J \sum_{i=1}^{J} U_i(\mathbf{d})$

end for

Choose the design $\mathbf{d}^*$ with the maximum utility $\hat{U}(\mathbf{d})$

mally differentiates one chosen model from many. Two questions arise here that have, so far, been largely ignored: what constitutes a *design*, and is *maximum differentiation* the only criteria for design optimization? I will discuss each of these questions in turn.

In the only other account of design optimization for bandit problems, the only

elements of the design which are optimized are the reward probabilities for each arm (Zhang & Lee, 2010b). In the conclusion of their paper on *Optimal experimental design for a class of bandit problems*, Zhang and Lee (2010b) note that an obvious extension to the framework would consider additional elements of bandit problems as part of the design optimization process, elements such as the number of arms and the number of trials in a problem. One goal of this project was to consider a wider range of bandit problem designs than have been considered so far, and so when developing the grid search algorithm for this project, I expanded the space of possible designs, optimizing not only for reward probabilities but also the number of arms and number of trials in a problem. In section 4.3.3 I will discuss another extension to this framework which addresses the problem of how experiments of a fixed number of trials should be divided into N separate bandit problems.

The next question is whether *maximum differentiation* is the only criterion for design optimization? A key consideration for experimentation in psychology is to design experiments that provide clear evidence for and against competing models; this is particularly useful in experiments where multiple models are considered, and researchers are interested in designing experiments that can effectively distinguish them (Cavagnaro, Myung, Pitt, & Kujala, 2010). Having an experiment that maximally differentiates between models allows researchers to draw stronger conclusions about the models people use. I believe that the optimal experimental design framework can be extended so that it can be applied not only to type of experiments described above, but also those that seek to test a particular hypothesis concerning the testing of a particular model. I believe that this can be done through the design of the local and global utility functions.

Karl Popper, a renowned Philosopher of Science, remarked that science seeks to disconfirm, and that confirmations should only count if they come from risky predictions, i.e., those that seek to destroy a theory; he stated that the only good test of a theory is one that is trying to falsify it (Popper, 2014). The hypothesis that has motivated this project is that the majority of people use the e-WSLS method as their dominant decision-making strategy when solving bandit problems. In order to gather evidence in favor of this hypothesis, I should, therefore, seek to disprove it. One way to do this would be to design an experiment that pushes people away from the use of the e-WSLS model, and towards alternate decision-making strategies.

The idea of controlling the decision-making strategy that people use by manipulating experimental conditions has been explored in the past, particularly in the domain of mathematical models in psychology. Luce (1995) drew attention to the difficulty of making sense of experimental data if careful attention is not paid to how this data may have been generated, i.e., by considering the multiple decision-making strategies that people may have used throughout an experiment. To alleviate this difficulty, he recommends improving both the ability to distinguish between different models, and controlling which model is used; the former has been discussed at length in this chapter, with attention now being turned to the latter. Such attempts to control a persons decision-making strategy have been exampled in the past: Wandell (1977) used descriptive models to identify and manipulate subject's strategies when measuring the speed-accuracy tradeoff in visual detection, and (Green & Luce, 1971) demonstrated that subjects could be induced to switch between 2 strategies given different payoffs on the task of auditory detection. We also see in the literature on bandit problems that people adapt their decision-making strategies based on the problem they are facing,

with one example being the increased tendency to exploit known, good, options when fewer trials are remaining (Christian & Griffiths, 2016). I, therefore, believe that it should be possible to manipulate subjects' decision-making strategies in bandit problems. In doing so, I will be better able to test the theory that the e-WSLS method is the dominant strategy people use when solving bandit problems. In summary, in order to falsify this theory, experiments would have to be designed so as to induce subjects not to use the e-WSLS method. I sought to do this by manipulating the utility functions defined in Equations 4.7 and 4.9.

Thus far, the utility functions described have been designed to maximize the badness of fit of a set of models on data generated by some chosen model, and vice versa, given a design $\mathbf{d}$. Additional criteria could be that the chosen model performs badly on that design. We can measure model performance on a given design by instantiating a model and having it "play" that design. The average score across all instantiations would inform us on how a model is expected to perform on that design. Last year I developed a series of Scala[3] classes which generated an artificial data set of decisions according to each of the models defined in Chapter 3 for the task of model identifiability (Laverty, 2019). Fortunately, when developing this code last year, I designed it for re-use: the classes took as inputs the parameters of a model and details of the bandit problem such as the number of arms and number of trials. I was, therefore, able to use this code to generate data for each model on each design in the grid of designs, D. Each model was instantiated with a given set of parameters, and "played" a given design, $\boldsymbol{d}$. After a fixed number of trials, the game ended and the total number of

[3]https://www.scala-lang.org/

rewards received by each model was recorded.

4.2.2 Design filtering

Now that we can calculate a model's performance on a single design, we can turn to the question of how to incorporate these results in our search for an optimal design. First, I will discuss what value we should derive from the absolute score a model achieves on design, and then I will discuss how this value should be incorporated into our utility functions.

There is no value in using the *absolute* score achieved by a model in calculating the optimality of a design; instead, to induce subjects to use one model over another, we should look at the *relative* performance of those models against some baseline. The baseline could be the total possible score that could be achieved by any model, i.e., the total number of rounds played where success is achieved on every round. This is, of course, unrealistic since it does not account for the reward rates of each arm. We could, instead, set the maximum possible score according to the maximum expected score which could be achieved by playing a given bandit perfectly. For example, the bandit problem with 2 arms with reward rates $(0.1, 0.9)$ has a maximum expected score of 90 when played for 100 trials, where on each trial, the second arm is chosen. This, too, is an unrealistic baseline for performance, since it assumes perfect knowledge of the distribution of rewards for a given bandit problem. We could simply calculate the expected score that one would achieve if they played the bandit randomly, and use this as a baseline. In order to push participants toward using other models I believe that rather than using the guessing model as a baseline we should use the maximum score that *any* model, from a fixed set of plausible models M, achieved on a given bandit

problem. Given a large enough set of plausible models, we can rank models within this set by computing the ratio of each models' score the maximum score achieved by any model in M. This project is concerned with testing whether humans primarily use the e-WSLS model when solving bandit problems. As such, I believe that it makes sense to design an experiment where this model performs poorly *relative to other decision-making strategies humans are believed to use*, so as to induce participants to use those other strategies. The score I have used to measure the performance of a model on a given design is, therefore, the ratio of that models score relative to the maximum score of any model of human decision-making on that bandit problem, where the scores of all models are calculated using the process described above.

Of course, we do not simply score "a model", but instead an instantiation of that model with a given set of parameters. The question then arises of what parameters we should use for a given model. One option is to compute the marginal score of a model on a given design by averaging over a grid of all possible parameter settings, computing the score the achieved at each setting. Another option would be to set the parameters of each model equal to perfect accuracy of execution since this reflects the "true" underlying model, not subject to any observational noise; this method has been adopted in the past when generating artificial data (Steyvers et al., 2009). Finally, since these scores will be used to design experiments that induce model use in human subjects, it makes sense to estimate the scores that humans are likely to attain by using each model. This can be done by generating multiple samples of model parameter values using priors obtained by fitting models to data from previous experiments of human decision-making on bandit problems, and averaging scores attained over these samples. Such prior values were obtained in the previous year of this project (Laverty,

2019; Steyvers et al., 2009), and so were used to draw such samples. One problem with this approach is that it rests on samples from a relatively small number of data points - 451 in this case - and from a single experiment. These priors on parameter values were validated against a second dataset in last year's project (Laverty, 2019; Zhang & Lee, 2010a), and were found to be largely consistent with previous results. As such, I had some confidence in proceeding with these values. Following the pseudo-code presented in (Zhang & Lee, 2010b) and the method outlined in Algorithm 1, the parameter sampling method was used.

The second question was how these values should be used to find optimal designs. One option would be to incorporate this value directly into the local utility function defined in Equation 4.7, modifying it like so:

$$u(\boldsymbol{d}, \boldsymbol{\theta}_m, \boldsymbol{y}_m) = \sum_{m' \in M, m' \neq m} BF_{m/m'} + \frac{s_{max}}{s_m} \tag{4.10}$$

where s_{max} denotes the score achieved by any model on the given design, and s_m is the score achieved by the chose model. When the global utility function is maximized according to Equation 4.9, bandit designs where the chosen model performed poorly relative to other models will be selected.

Another option would be to filter designs based on the ratio of $\frac{s_m}{s_{max}}$ before the utility of the design based on model discriminability is calculated, choosing designs where $\frac{s_m}{s_{max}}$ falls in some range $a \leq \frac{s_m}{s_{max}} \leq b$. The advantage of this approach is that we may specify the range of values the ratio of scores could take: when considering the approach of optimizing this ratio within the local utility function as is shown in Equation 4.10, I found that the designs with the largest value for this ratio - what we would optimize for - were ones where *both* the e-WSLS *and* best performing model performed terribly, where the best performing model achieved, for example, a score

of 5 in a bandit problem played for 100 trials, with the e-WSLS model scoring 0 or 1. As I wanted the participants in my experiment to be engaged, I wanted them to feel that they could achieve a satisfactory score. A low scoring design may degrade the competence a player would feel, and therefore affect their motivation to give their full effort in the experiment (Wang & Sun, 2011). I, therefore, decided that filtering designs manually, rather than optimizing automatically, was the right strategy for this project.

Another advantage is that we do not have to compute the costly local utility function for all designs. While there may be many designs where discriminability is high, we will not want to consider them if our chosen model does not perform poorly in them. This filtering of designs prior to optimization reduces the overall number of computations that must be conducted in our grid search. I do believe, however, that there is value in modifying the utility function directly, and it highlights the flexibility of framework proposed by Myung and Pitt (2009); perhaps this idea will be of use to future researchers testing the use of a specific model like I am here, and who opts for sampling strategies, rather than a grid search, in optimizing the global utility function.

In summary, in order find optimal designs of bandit problems to falsify the claim that e-WSLS is the dominant decision-making strategy used by humans when solving bandit problems, designs were filtered according to the score of the e-WSLS model relative to the maximum score attained by any heuristic cognitive model of human decision-making on bandit problems. These filtered designs were then subjected to the optimal experimental design framework proposed by Myung and Pitt (2009), adapted so as to maximally differentiate the e-WSLS model from all other cognitive models,

using the grid search showing in Algorithm 1 to estimate the global utility function in Equation 4.9.

4.2.3 Combining individual designs

Finally, all of the designs discussed so far were for a single bandit problem with a fixed set of reward rates and a given number of trials. In their discussion of future work to be carried out on optimizing designs for bandit problems, Zhang and Lee (2010b) note that "One particularly interesting problem would be to assume that participants are available for a fixed number of trials, and optimize how those trials are divided into problems". They mention that it is not immediately obvious whether an experiment should consist of many bandit problems, each played for only a few trials, or whether experiments with only a few problems, played for a large number of trials, is optimal. The extensions was also considered in the design optimization framework of this project.

While Algorithm 1 and Equation 4.9 are concerned with finding a single optimal design, they can be adapted to produce a set of N of high-quality designs centered around this optimum. A sample of n designs from the set N can then be combined and evaluated according to the criteria discussed for single bandit problems discussed so far in this chapter, producing an optimum $game$[4]. This extension to the optimal experimental design framework for bandit problems is not only of theoretical interest - answering the question of the tradeoff between the number of problems and number of trials introduced above - but it is imperative to this project as it is ultimately the performance of people on a series of bandit problems which will be evaluated.

[4]where a game consists of n individual bandit problems

In their application of the DO framework proposed by Myung and Pitt (2009), Zhang and Lee (2010b) also focused on producing a series of high utility designs, rather than a single optimum design, as I have adopted here. One reason they give for this is that by considering many high utility designs, they can give some variability and robustness to the experiments created using these designs. The tradeoff between variability and utility arose in the final stages of design in this project. In order to deal with it, the sampling procedure was modified to generate games from bandit problems so that each game would have an even distribution of reward rates across all arms, averaged over the entire game. This reduced the likelihood that there would be any meta-strategy that could be used to exploit the structure of the game. This modification has likely produced a game design which has a lower utility score than the optimum solution produced by solving Equation 4.9 at the cost of increased variability. This example highlights the need for human intuition and intervention in experimental design which, at least for now, cannot be quantified and optimized.

Discussion so far has been shy of technical details, such as how the computational inefficiency of the grid search was overcome, as well as results indicating what form optimal designs would take. This has been purposeful so as to separate the theoretical details of the OED framework from the implementation and results; those details will be presented now.

4.3 Implementation and Results

The outline of this section is as follows: I will begin with a discussion of exactly what elements of a bandit problem I quantified and optimized for in each design. Next, I will discuss the implementation of the grid search method to filter these designs according

to the criteria outlined in Section 4.2.2. I will also provide results summarising the estimated utility of various types of designs. Finally, I will conclude with a brief description of how high utility designs were combined to create a game. I will present a few results of this framework highlighting the utility-variability tradeoff discussed in the previous section, before finally presenting the final design I chose to use in my experiment.

4.3.1 Design components

As noted in the section introducing the grid search method, the designs I optimized for included: the number of arms in each bandit problem, the number of trials the problem should be played for, and the distribution of rewards across arms.

All possible combinations of these parameters were considered as the search space for possible designs. The range of values for each parameter was as follows: for the number of arms, $[2, 3, 4, 5]$; for the number of trials, $[5, ..., 50]$, in steps of 5; for reward rates for each arm $[0.0, ..., 1.0]$ in increments of 0.05. For each design with n arms all permutations of length n were created from the set of possible reward rates. Each model "played" each of these designs in order to generate a value use for design filtering; details of this work are presented next.

4.3.2 Grid search

Other researchers have shied away from the use of a grid search to solve the OED problem, opting for the use of statistical methods instead, due to it's computationally expensive nature. While previous studies used sampling methods when the design space considered only 2-arm bandit problems (Zhang & Lee, 2010b), I was able to use

a grid search method to solve the OED problem up-to-and-including five arms. This was made possible by implementing the pseudo-code of Algorithm 1 in Scala along with the *Apache Spark*[5] framework to enable the grid search to be carried out on the Eddie computer cluster (*ECDF*, n.d.). Initial scripts to enable the use of Spark on Eddie were based on a tutorial created for such a task[6]. By utilising the parellisation and distribution enabled by Spark, as well as the large number of compute nodes made available by the Eddie cluster, the grid search could be conducted within the time frame of this project.

The first pass of design filtering involved filtering designs where the e-WSLS model was in the bottom 3[7] scoring models for a given design, relative to the best scoring model. Figure 4.1 shows the average ratio of the score achieved by the e-WSLS model to the maximum score achieved by any model across all designs for a given number of arms and number of trials. We can see from Figure 4.1 that the number of arms in a design has a much greater effect on the ratio of scores than the number of trials does - the average ratio is almost constant across all number of trials for all number of arms except 4, where the average ratio decreases as the number of trials increases. We see that the lowest average value - indicating the design which the e-WSLS performs worst in, and likely the design which would induce participants to use an alternative strategy - is a 5-armed bandit played for 20 trials.

In addition to knowing the ratio of the e-WSLS score relative to the best scoring model, we also want a sense of the expected score that is achievable for a given de-

[5]https://spark.apache.org/

[6]https://github.com/rosafilgueira/Spark_EDDIE_TextMining

[7]Being in the bottom 3 was used as a filtering criterion as almost all designs had the guessing model as the worst scoring model, and the WSLS model scored almost identically to the e-WSLS variant.

sign - as discussed earlier in this report, low scoring problems may limit participants' engagement on the task. The average ratio of scores achieved by any model to the number of trials (taken as a proxy for the maximum possible score, rather than calculating expected scores, for simplicity) is shown in Figure 4.2. We can see from Figure 4.2 that as the number of trials and number of arms increases, the best-scoring model achieves a higher overall score, relative to the number of trials the problem is played for. Combining the results of Figures 4.1 and 4.2, we see that the initial design filtering was successful in producing bandit designs where e-WSLS performs poorly relative to other models based on the scores achieved by these models alone.

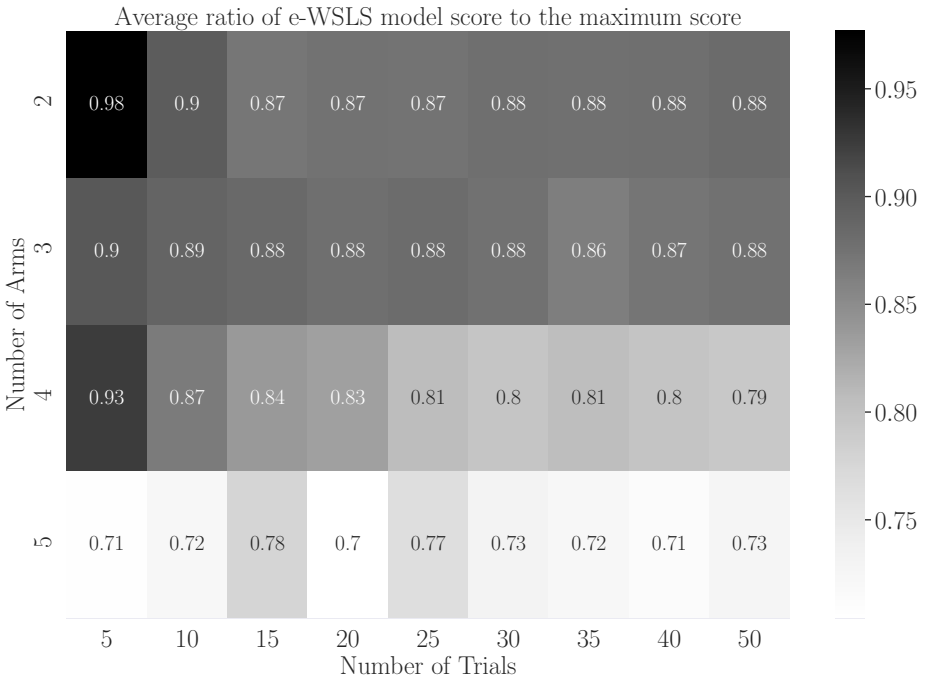


Figure 4.1: Average ratio of e-WSLS model score to the maximum score, across bandit designs with different numbers of arms and trials

Of course, designs in which the e-WSLS model performed poorly are only one part of the solution to finding OEDs for this project, with the other component being the

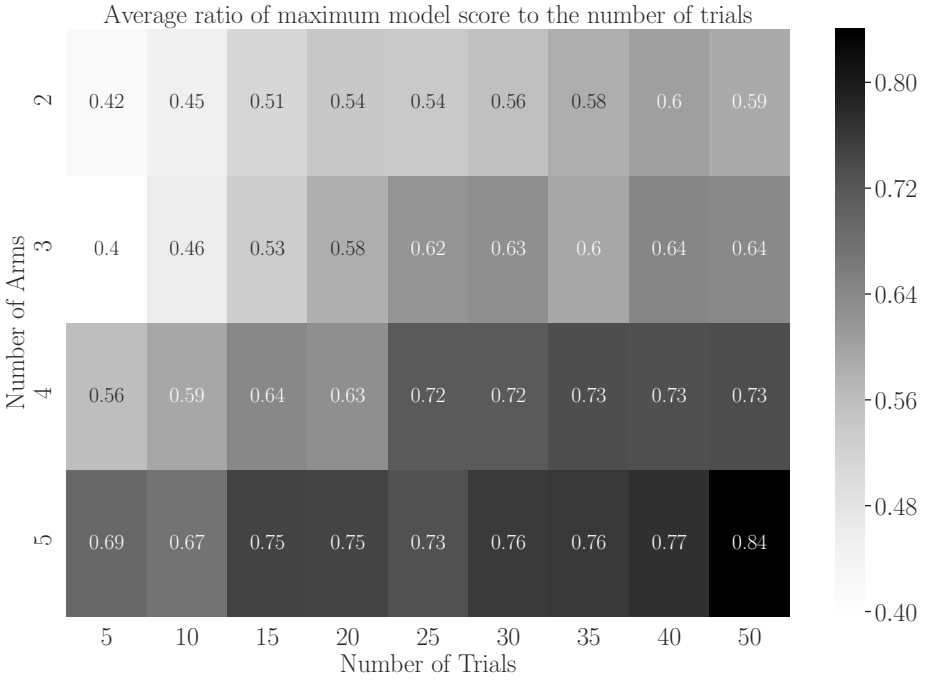


Figure 4.2: Average ratio of maximum model score to the number of trials, across bandit designs with different numbers of arms and trials

global utility of those designs, i.e., the ability of a design to maximally differentiate between the e-WSLS model and all others. The average global utility of designs is shown in Figure 4.3[8]. What is interesting to note from Figure 4.3 is that the number of arms in a bandit problem does not seem to have much of an effect on the average global utility value, with similar utility values calculated across all arms for a given number of trials[9]. The effect of the number of trials on the utility value, however, is obvious: increasing the number of trials increases the global utility of that design. This result makes sense since as the number of trials increase, more information is

[8]No results are shown for 2-armed bandits played for 5 or 10 trials since design filtering removed all of these designs

[9]Note that the utility values take on negative values here. This is because in practice I worked with log values to avoid numerical issues. Where I mention maximising global utility therefore, in practice I was actually minimizing the negative log of the global utility.

available to be put to use in distinguishing one model from another.

Increasing the number of trials increases the global utility of a design, however, what happens when the total number of trials is fixed, and an experiment is broken up into different numbers of games at different trial lengths? To answer that question, we can turn to the analysis of the game design.

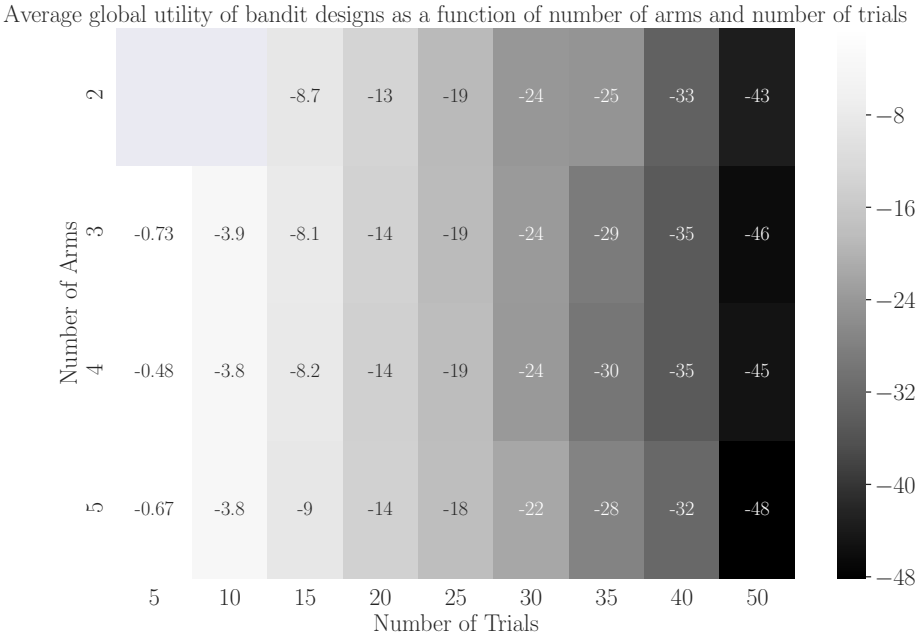


Figure 4.3: Average global utility of bandit designs as a function of number of arms and number of trials

4.3.3 Game design

To generate a complete experiment consisting of many bandit problems, I took the top 1000 bandit designs produced by Algorithm 1 for a given number of arms and trials, and sampled n bandit problems from this set - the actual value of n depended on the number of trials in each problem, so that the total number of trials was about 300, a guideline for experimental design I have adopted from numerous studies (Steyvers et al., 2009; Zhang & Lee, 2010b). The sampling procedure was repeated 10,000 times

to ensure that each design was likely to have been sampled. The global utility of the constructed games was re-calculated using Algorithm 1, which was adapted to compute the global utility of a game, rather than individual bandit problems.

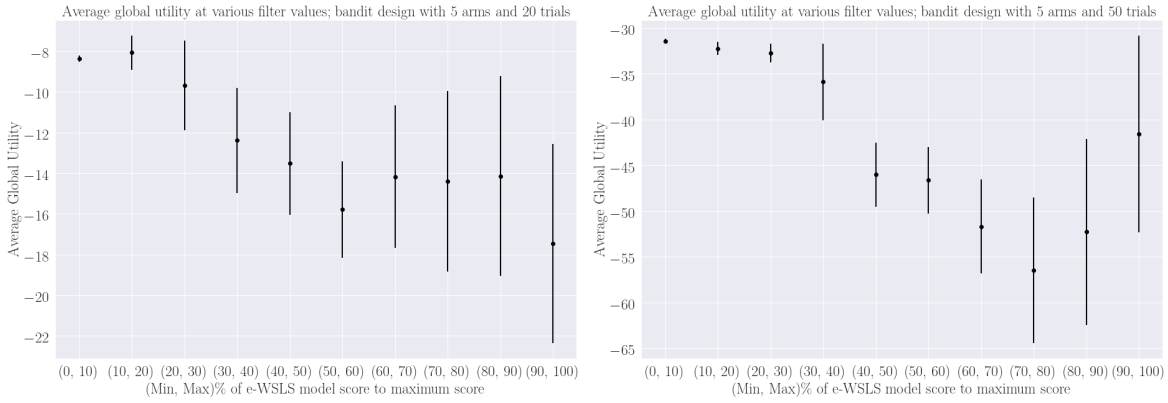
The motivation for this analysis was two-fold: first, by fixing the total number of trials to 300 and varying the number of bandit problems within a game, the question as to whether games with a large number of shorter problems have a higher utility than a game with few problems each played for a large number of trials can be investigated; secondly, many possible game designs can be sampled and evaluated, ultimately producing a high-utility game design which can be used in an experiment with human participants.

One note before proceeding here. Figure 4.3 shows the *average* global utility of a design, while Figures 4.1 and 4.2 show the *average* ratio of scores achieved by models on a design. These values are averaged over all reward distributions for a given number of arms and number of trials. It is unwise to sample bandit problems based on these average values alone: we may end up sampling 5-armed bandits played for 20 trials where the ratio of the e-WSLS model's score was closer to 100% of the maximum model score, rather than the 70% average indicated in Figure 4.1. The same caution should be taken when drawing conclusions based on the average utility values shown in Figure 4.3. Figure 4.4, therefore, shows the average utility - along with error bars showing the standard deviation - across all bandit designs for 5-armed bandits played for either 20 (Figure 4.4a) or 50 (Figure 4.4b) trials. The designs are split based on the average ratio of the score of the e-WSLS model to the maximum score achieved by any model. We can now see a large variation in the utility of bandit designs for a fixed number of arms and trials: Figure 4.4b, for example, shows utility values ranging from

around -30 to -65. The best filter value for 20-trial problems seems to be $50-60$, while for 50-trial problems, it is $70-80$. On balance, however, a filter value of $70-80$ seems quite high if one of the goals of DO is to find designs where e-WSLS performs poorly relative to other models. A filter value of $50-60$ was, therefore, applied to both designs so that this criterion was satisfied for all bandit problems; this is the optimal setting for 20-trial bandit problems, and yields an average global utility of -46.6, a still very high utility value. This example highlights the benefit of the design filtering technique, as we can make decisions on the threshold values used for filtering on a case-by-case basis.

These two sets of designs were then used to generate games to investigate the tradeoff between the number of trials and the number of problems in a given experiment. These designs were used since results from Figures 4.1, 4.2, and 4.3 show that these particular designs exhibited a good tradeoff between a low relative score for the e-WSLS model and a high global utility. The number of trials in each design is also different by greater than a factor of 2, making them ideal candidates for investigating the number-of-trials vs. number-of-problems tradeoff. A more complete experiment would consider all possible combinations of problem and trial lengths that resulted in a game lasting for 300 trials, however, such an experiment was beyond the scope of this project.

The average global utility of a game constructed by sampling from each set of designs was $-132.8(2.85)$ and $-141.3(5.76)$ for the 15-problem 20-trial game, and the 6-problem 50-trial variants respectively - note how the total number of trials is fixed at 300 in both games. This result indicates that perhaps the tradeoff between number-of-problems and number-of-trials may not be an important one to optimize for when



(a) 20 trials

(b) 50 trials

Figure 4.4: Mean global utility of 5-armed bandits at different average ratios of e-WSLS scores to the maximum model score

designing experiments with bandit problems. Perhaps as is indicated by Figure 4.3, the total number of trials is what will have the largest effect on the utility of an experiment. Of course, this is only one piece of evidence, and to draw and concrete conclusions on the nature of the tradeoff - which is beyond the scope of this project - more combinations of number-of-trials and number-of-problems must be evaluated. Following the results of this brief experiment, I decided to proceed with the 20-trial problems as I believed that this setup would be more beneficial for testing my per-game e-MM: both experiments have the exact same number of trials so the per-trial mixture models would be unaffected whereas the per-game model would be limited in an experiment with only 6 problems.

Finally, the issue of the utility vs. variability was to be dealt with. An example of the highest utility design constructed from 15, 20-trial, 5-armed bandit problems is shown in Table 4.1a.

We see that there is almost no variation in the distribution of rewards across arms. It is likely that a human playing the game shown in Table 4.1a would recognize the

Arm 1	Arm 2	Arm 3	Arm 4	Arm 5
0.45	0.35	0.85	0.5	0.45
0.55	0.2	0.85	0.55	0.4
0.55	0.0	0.85	0.5	0.5
0.55	0.15	0.85	0.55	0.45
0.55	0.3	0.85	0.5	0.45
0.45	0.15	0.85	0.55	0.5
0.55	0.25	0.85	0.5	0.4
0.55	0.25	0.8	0.6	0.45
0.55	0.2	0.85	0.5	0.5
0.55	0.1	0.75	0.55	0.4
0.45	0.2	0.8	0.5	0.45
0.45	0.2	0.85	0.5	0.45
0.55	0.1	0.8	0.55	0.45
0.45	0.25	0.8	0.5	0.45
0.55	0.1	0.85	0.5	0.45

(a) Original optimal game design

Arm 1	Arm 2	Arm 3	Arm 4	Arm 5
0.2	0.05	0.3	0.15	0.6
0.35	0.0	0.0	0.7	0.2
0.2	0.65	0.25	0.4	0.2
0.2	0.0	0.15	0.15	0.6
0.1	0.05	0.6	0.1	0.6
0.3	0.65	0.25	0.35	0.2
0.6	0.0	0.3	0.15	0.0
0.0	0.05	0.6	0.15	0.6
0.2	0.65	0.25	0.4	0.15
0.35	0.25	0.05	0.7	0.1
0.6	0.05	0.3	0.05	0.0
0.05	0.0	0.6	0.1	0.6
0.3	0.0	0.2	0.25	0.6
0.6	0.05	0.3	0.15	0.0
0.35	0.15	0.05	0.7	0.15

(b) Modified optimal game design

Table 4.1: Reward distributions for arms produced by original and modified versions of the optimal game design framework. Games created using samples from 5-armed bandits played for 20 trials.

pattern of high rewards for Arm 3, and perhaps stick with that arm throughout the duration of the game.

The game sampling procedure was, therefore, modified to impose the restriction that

there be a wider distribution of rewards across arms; this modification involved ensuring that the average expected rewards should be roughly the same for all arms. A batch of $10,000$ samples of games, using the same set of bandit problems originally used, was generated and evaluated; the design with the highest utility is shown in Table 4.1b. A larger variation in reward distributions can be seen in this modified design; however, the resulting utility of that design is -104.6 - where the global utility of the design in Table 4.1a is -141.2.

These cases exemplify the utility-variability tradeoff discussed throughout this chapter and highlights the importance of human intervention in experimental design, rather than strict reliance on utility values. Regardless of the lower utility of the modified design, this design was used in the experiment conducted for this project, with variability outweighing utility in this case.

4.4 Experiment

4.4.1 Method

With an optimal design identified, the next step in the project was to design an experiment in which it could be used.

A simple interface was developed in order to allow participants to solve the bandit problems; an example of this interface, captured in the middle of a game, is shown in Figure 4.5. The design used for the experiment was exactly the one shown in Table 4.1b: 15 5-armed bandit problems, each played for 20 trials. The same seed was used to generate rewards according to the distributions shown in Table 4.1b for each participant, so that observed variations in strategies used between participants were

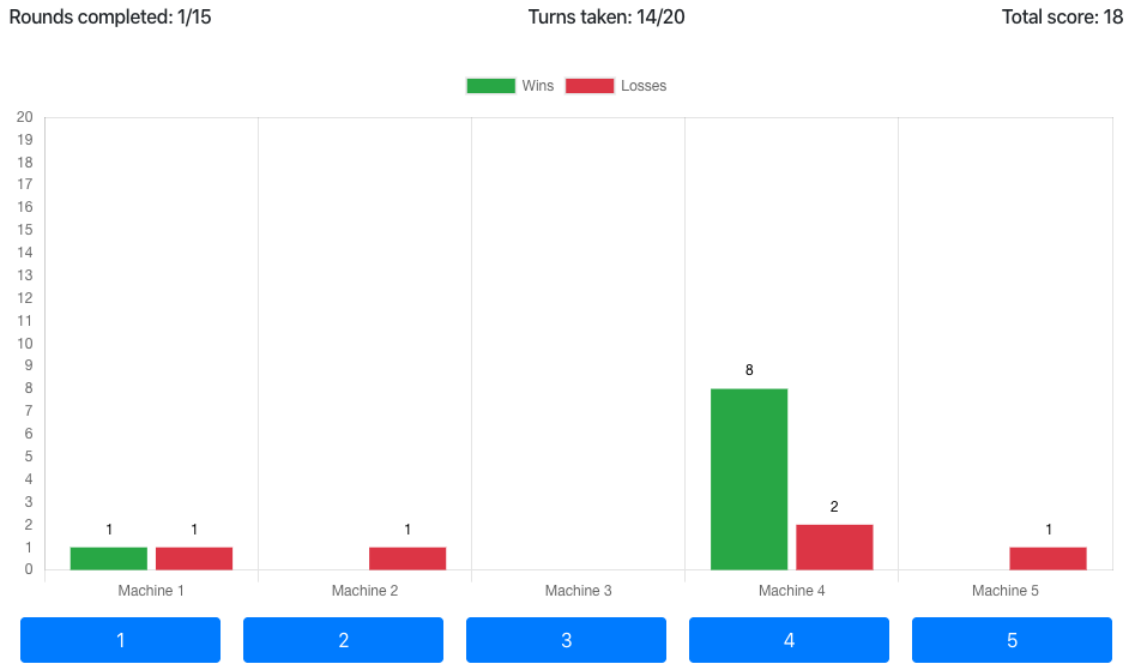


Figure 4.5: Screenshot of experiment interface

not due to random variations in rewards received.

Playing the game was straightforward, and the following preamble was presented to participants prior to beginning, to ensure they understood how to play:

> *Below, you can see a number of options. Think of each of these options as an individual slot machine. Like real slot machines, you press a button (or pull a lever) and you either win, or you lose. Each of the machines below has a different payoff rate: the number of wins or losses you can expect from that machine. Some machines will give more wins than others, while some may give losses most of the time (think of each button press as flipping a coin, where each machine has a different probability of giving you a head or tails, and each button press is independent, even for the same machine). On each round of the game, you'll have 20 turns on the current set of machines. On any turn you can choose any machine, and you'll be*

told immediately whether that turn was a win or a lose, and this will be added to your total score: a win will count for 1 point, and a loss will earn you 0 points. During a single round, the payoff rate of each machine will stay the same. At the end of the current round the machines will reset: the wins and losses on each machine will be set to 0, and the payoff rate of the machines will change for the next round. Your goal is to maximize your total overall score across all 15 rounds. Good luck!

Figure 4.5 shows the experimental interface in the middle of a game: so far the participant has completed 1/15 problems and has played 14/20 trials on the current problem resulting in 1 success and 1 failure for Arm 1, a failure on Arm 2, 8 successes and 2 failures on Arm 4, and a failure on Arm 5. Arm 3 has not been chosen. Over the course of this game, the participant has amassed 18 points.

4.4.2 Implementation

The game was developed from scratch using a MERN stack:

- MongoDB[10] - specifically the MongoDB cloud store Atlas[11] - was used as the database to store participant information including their consent, a unique ID for each participant, and their results from playing the game
- Express[12] was used to handle server requests
- React[13] was used to develop the actual interface with which participants interacted

[10]https://www.mongodb.com/

[11]https://www.mongodb.com/cloud/atlas

[12]https://expressjs.com/

[13]https://reactjs.org/

- finally, Node.js[14] was the serving framework upon which Express was built on top of

The resulting application was hosted on the cloud platforming service Heroku[15].

4.4.3 Participants

A total of 138 participants completed the bandit problems developed for this project. Participants were recruited via two methods: a mass email was sent to the *students@inf.ed.ac.uk* address recruiting all students within the School of Informatics at the University of Edinburgh; the second method involved recruiting workers on Amazon Mechanical Turk[16]. The first method recruited 99 participants, with the remaining 39 coming from the Amazon Mechanical Turk service.

The experiment was reviewed and certified according to the Informatics Research Ethics Process, RT number 4355. All participants were presented with a participant information sheet and consent was obtained prior to beginning the experiment. The results of the experiment will now be presented.

[14]https://nodejs.org/en/

[15]https://www.heroku.com/

[16]https://www.mturk.com/

Chapter 5

Analysis

This chapter describes the results of the experiment described at the end of Chapter 4. The Chapter will proceed as follows: Section 5.1 will begin with details of a brief artificial recovery study I conducted using the optimal experimental design (OED) in order to test model identifiability. Next, I will describe a study of individual differences in model choice and parameter values observed after applying the models described in Chapter 3 to the collected experimental data, referred to from here on out as the *OED dataset*. Section 5.2 describes individual differences in model use and parameter values obtained via an analysis of hierarchical Bayesian models applied to the OED dataset, with results of this analysis compared to findings of a similar analysis conducted in last years project on the testweek dataset described in Chapter 2. In that section, I will also describe the characterization ability of each model on human and optimal decision making data via a posterior predictive agreement analysis, noting which model best describes each dataset. This chapter will conclude with a comparison of posterior estimates of the psychological parameters obtained by fitting each model to both human and optimal data sets, drawing conclusions on

human versus optimal performance.

5.1 Non-Hierarchical Models

5.1.1 Model Identifiability

Model identifiability and parameter recovery are important, especially for cognitive models where the interpretation of model use and parameter values as models which are not identifiable, and parameters which are not well recovered are of limited value (Farrell & Lewandowsky, 2018). I, therefore, decided to begin my analysis with a test of model identifiability and parameter recovery in order to ensure that the models considered here would be useful on this particular bandit problem.

To carry out this analysis, I made use of the Scala classes discussed in Chapter 4. The same classes were used in a similar analysis of the testweek dataset conducted last year. The results using the implemented Scala classes indicated almost perfect model identifiability and parameter recovery. I was, therefore, confident in the correctness of the implementation of those model classes, and so felt assured in re-using them for this year's analysis. While it may seem counterproductive to repeat the model identifiability study since I am using the same models and the same implementations which a previous study suggested were correct, that study only tested model identifiability and parameter recovery for that particular set of bandit problems; model identifiability is tied to the dataset used to conduct the analysis, and since I am using a new set of bandit problems this year, I needed to re-test model identifiability and parameter recovery.

Before describing the results of this study, I should note that last year's project ex-

cluded the analysis of the Optimal model described in Chapter 3 on the artificial data sets as the likelihood function for the Optimal was too computationally expensive to compute; computing the likelihood function for a single participant's data took eight hours[1] when parallelized across an eight-core machine (Laverty, 2019). Given the fact that this year's bandit problems feature the additional complexity of an additional arm, including the Optimal model in the analysis was simply not feasible given the time constraints of the project, and so it is once again omitted.

To conduct the analysis, an artificial dataset was generated for each model. The dataset consisted of simulated decisions made for 1000 "participants" - each "participant" is an independent sample, where each sampled used a different seed to generate rewards for each arm, allowing for an analysis of how each model would perform under different reward scenarios. Each "participant" completed the given set of bandit problems, solving the bandit problem according to that particular model. For each artificial dataset, the parameters of the generating model were set to the perfect accuracy of execution, as is the case in previous in artificial recovery studies (Steyvers et al., 2009). Once each dataset was generated, the marginal likelihood of that dataset under each model was calculated. The Bayes factor for each model relative to the guessing model was calculated using these marginal densities as shown in Equation 5.1. To approximate the integral shown in the numerator of Equation 5.1, the likelihood of the data under a given model was calculated over a grid of parameter values

[1] clock time

for that model.[2]

$$BF_{A/Guess} = \frac{\int p_A(D|\boldsymbol{\theta}_A)p(\boldsymbol{\theta}_A)\, d\boldsymbol{\theta}_A}{p(D|Guess)} \tag{5.1}$$

The log of the Bayes factor - henceforth referred to as log BF for conciseness - for each model relative to the Guessing model are shown in Table 5.1 and Figure 5.1. What I should clarify here is that in last year's model identifiability study I stated that I was analyzing the "ability of the log BF measure to identify the underlying decision model". While others have attempted to use simulated recovery studies to evaluate methods of inference themselves (Pitt, Myung, & Zhang, 2002), the validity of such an approach has been questioned (M. D. Lee, Gluck, & Walsh, 2019). I should make clear that I was not, and this year that I am not, attempting to make any broad claims about the ability or validity of the log BF measure to uncover ground truth. The implication that I was trying to do so in last year's project was a product of incomplete understanding and confusing language use on my part[3]. Instead, what I am attempting to do - and all that I can do given the nature of simulated recovery studies - is to make comments on the accuracy of model implementation and the informativeness of the experimental design (M. D. Lee, 2018; M. D. Lee et al., 2019). The results here should, therefore, be taken with a pinch of salt. In essence, they are useful for showing the relationship between the models on this given dataset, not for drawing conclusions about the ground truth of model use and parameter values since

[2]In all models where it was appropriate, this involved a grid of 40 evenly spaced values over the domain $(0, 1)$ for the accuracy of execution parameters. Values of parameters such as τ and π were evaluated over an appropriate domain, e.g., *(1, Number of trials)*.

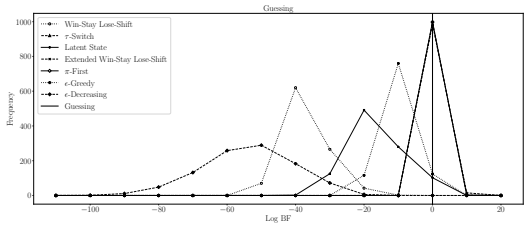
[3]Thank you to Michael Lee for drawing attention to this confusion, and for providing me with the resources to better understand the problem. Interested readers should see the appendix of (M. D. Lee et al., 2019) for further clarification of the benefits and limitations of recovery studies

this can change depending on the model selection criteria used (Pitt et al., 2002). These broader claims are better supported by analysis of Bayesian models (M. D. Lee et al., 2019), which are the subject of the next section of this chapter.

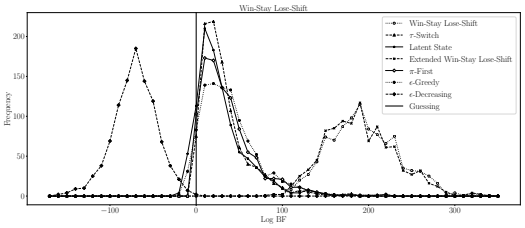
Recovery Model	Guessing	WSLS	e-WSLS	ϵ-Greedy	ϵ-Decreasing	π-first	Latent State	τ-switch
Guessing	76	0	1	0	0	4	1	17
WSLS	0	100	0	0	0	0	0	0
e-WSLS	0	100	0	0	0	0	0	0
ϵ-Greedy	0	0	0	12	0	86	1	0
ϵ-Decreasing	76	0	17	0	1	2	2	2
π-First	0	0	0	0	0	100	0	0
Latent State	0	17	3	14	0	0	47	18
τ-Switch	0	0	33	0	0	0	34	33

Table 5.1: Model recovery on artificial data (rows indicate the model used to generate the dataset

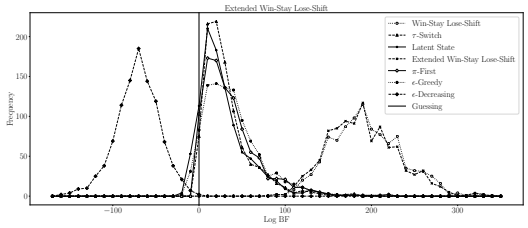
Table 5.1 and Figure 5.1 shows that model recovery on this particular dataset using the log BF measure aren't very promising. Only in two out of eight cases is the correct underlying model - the one which generated the data - identified as having the largest log BF value in 100% of samples; the WSLS and π-First models. The correct model is identified in the majority of samples for the Guessing and Latent State models; however, recovery is not perfect, with the Latent State model only being correctly



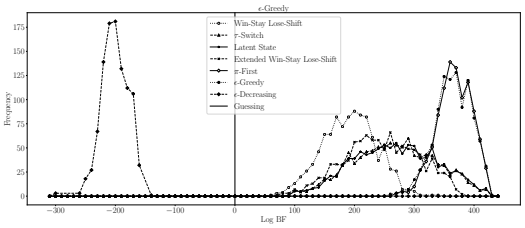
(a) Guessing



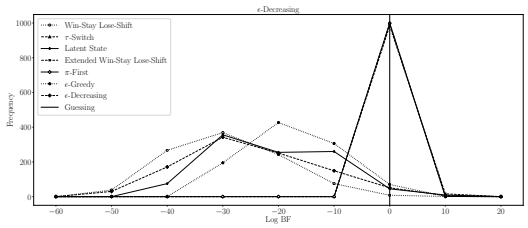
(b) WSLS



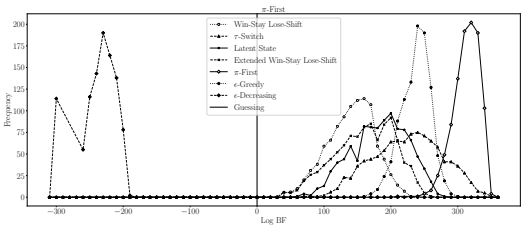
(c) e-WSLS



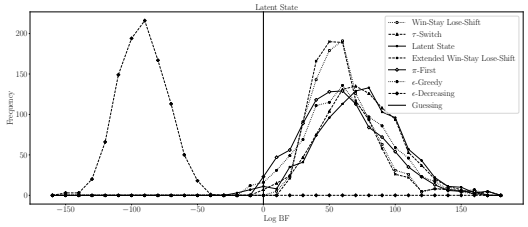
(d) ϵ-greedy



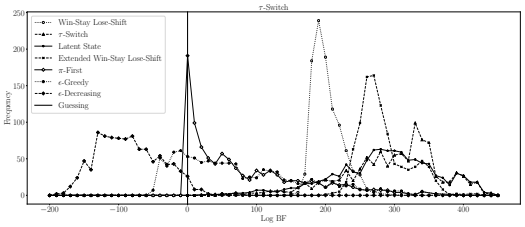
(e) ϵ-decreasing



(f) π-first



(g) Latent State



(h) τ-switch

Figure 5.1: Log BF measures for models applied to artificial datasets

identified in 47% of samples. There seem to be 4 cases where model recovery has failed: WSLS vs. e-WSLS; Latent State vs. τ-Switch; ϵ-greedy vs. π-First; and ϵ-Decreasing vs. Guessing.

I won't spend too long here discussing the reasons for the first two cases, as similar errors arose during the model identifiability study conducted in the previous year of this project (Laverty, 2019, p. 78-82), with an in-depth analysis provided for the

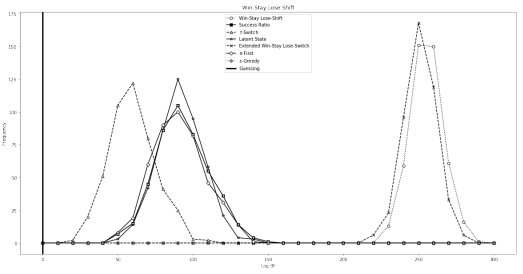
possible reasons for each case. What I will say, briefly, is that the confusion in these particular cases is not too surprising as in each pair of models one is a more complex, and therefore more flexible, version of the other: e-WSLS is simply the WSLS model with an additional parameter; and τ-Switch is a simplification of the Latent State model. The similarity of the models is also evidenced in Figure 5.1: Figures 5.1b and 5.1c show almost identical log BF measures for the WSLS and e-WSLS models, and Figures 5.1g and 5.1h the same can be said for the Latent State and τ-Switch models. Due to the similarity in each pair of models, the error in model identifiability can be justified[4]. I believe a similar argument holds for the large proportion of samples generated by the ϵ-Greedy model being attributed to the π-First model. The ϵ-Greedy model is a limited version of π-First model, and so the increased flexibility of the π-First model may have resulted in a better fit, and larger log BF value, than the ϵ-Greedy model. The similarity in log BF values between ϵ-Greedy and π-First models can be seen in Figure 5.1d, showing that while the π-First model had the larger log BF value in the majority of cases - as is seen in Table 5.1 - the differences between those values were only marginally bigger than the log BF scores of the ϵ-Greedy model.

Finally, turning to the large proportion of samples generated by the ϵ-Decreasing model being attributed to the Guessing model. We can see from Figure 5.1e that all models gave a poor fit to the ϵ-Decreasing data, and since Bayes factors were calculated with relative to the Guessing model it is clear why that model had the largest log BF value, even if it was 0. What I am unsure of, however, is why the ϵ-Decreasing model was not well recovered on this dataset. I can only speculate that

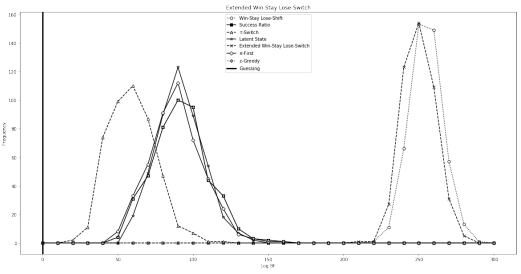
[4]Interested readers should see (Laverty, 2019, p. 78-82) for a discussion of how these errors are likely to have arisen during as a result of the marginal density calculations in producing the Bayes factors.

perhaps the ϵ-Decreasing model is not well suited to this bandit problem: we can see in every sub-figure of Figure 5.1 that the ϵ-Decreasing model had a log BF value less than 0, indicating more support for the Guessing model than the ϵ-Decreasing model across all artificial datasets.

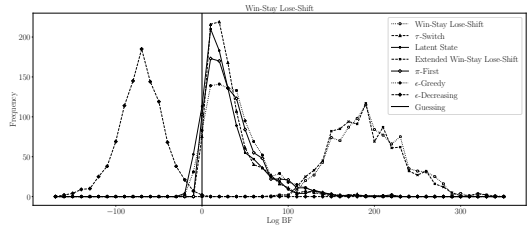
What is of concern in these results is that on the simulated data generated by the e-WSLS model, 100% of samples were believed to have been generated by the WSLS model. While I have presented a possible explanation for this result, it is still concerning that the e-WSLS was not well differentiated from all other models on its "own" artificial dataset. Shown in Figures 5.2a and 5.2b are results from a similar model recovery study conducted in the first year of this project. We can see from both figures that the log BF values of the WSLS and e-WSLS are almost identical, with WSLS slightly larger in both cases than e-WSLS, as is the case with this year's results. What is of note, however, is that in comparing Figures 5.2a and 5.2c, and



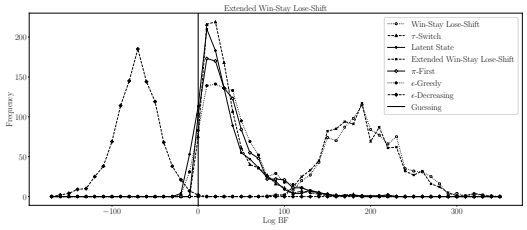
(a) WSLS (Laverty, 2019)



(b) e-WSLS (Laverty, 2019)



(c) WSLS

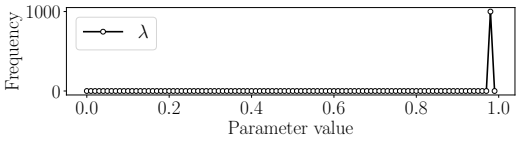


(d) e-WSLS

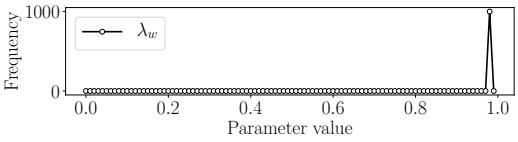
Figure 5.2: Log BF measures for WSLS and e-WSLS models on 2 different datasets

Figures 5.2b and 5.2d, that results from this year show less support for all other models on both the WSLS and e-WSLS artificial datasets; the mean difference in log BF values has increased from ~ 150 to ~ 180. The gap between the WSLS and e-WSLS models is also smaller in this year's design, indicating larger support for the e-WSLS model in both datasets. While it may be difficult to identify exactly which variant of the WSLS model was used, the gap between both variants and all other models when using a design produced by the DO framework is much larger. This result indicates a success of the DO framework to produce designs in which the e-WSLS model, or a variant of it, is maximally distinguished from all other models.

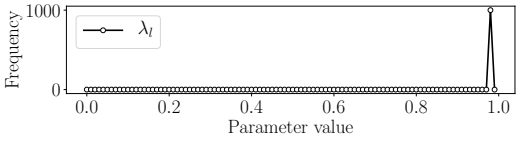
Moving on from model identifiability to parameter recovery, we can see from Figure 5.3, which shows the distribution of maximum a posteriori (MAP) estimates of the accuracy of execution parameters of each model, that parameter recovery was perfect for almost all models; the only exceptions are the parameters for the τ-Switch and Latent State models. Similar results for those models were also reported in last year's project (Laverty, 2019, p. 82). I believed in hindsight that last year's results were due to the errors in the implementation of the likelihood functions of these models, as was described in Chapter 2. However, we can see from Figure 5.3 that results are not much better this year after correcting this error. I double-checked the implementation of both models in case I changed anything when correcting the error mentioned above, as well as identifying any other errors I may have made when implementing the models. I could not see any obvious errors; of course, this isn't to say that they do not exist. I must, therefore, carry on under the assumption that the models are implemented correctly. Given the weak parameter recovery and model identifiability results for the



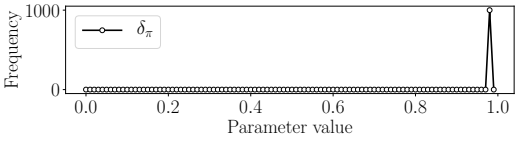
(a) λ (WSLS)



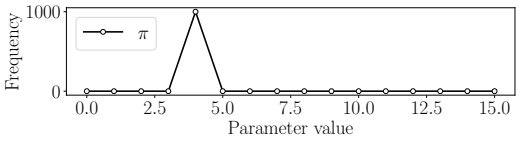
(b) λ_w (e-WSLS)



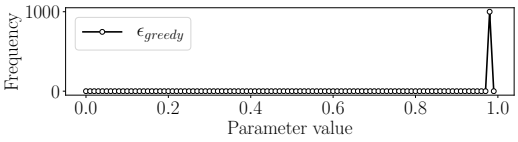
(c) λ_l (e-WSLS)



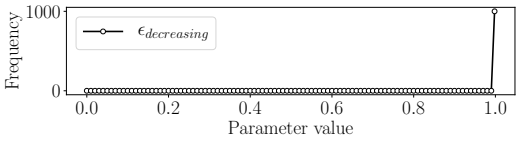
(d) δ (π-first)



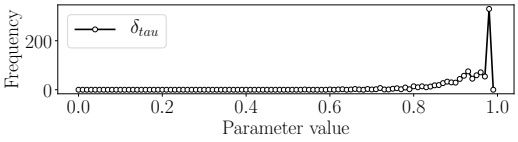
(e) π (π-first)



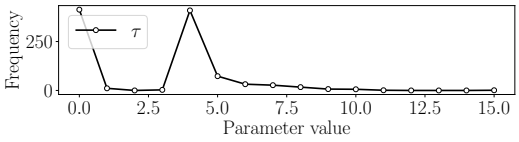
(f) ϵ (ϵ-greedy)



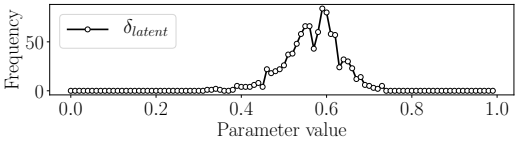
(g) ϵ (ϵ-decreasing)



(h) δ (τ-switch)



(i) τ (τ-switch)



(j) δ (Latent State)

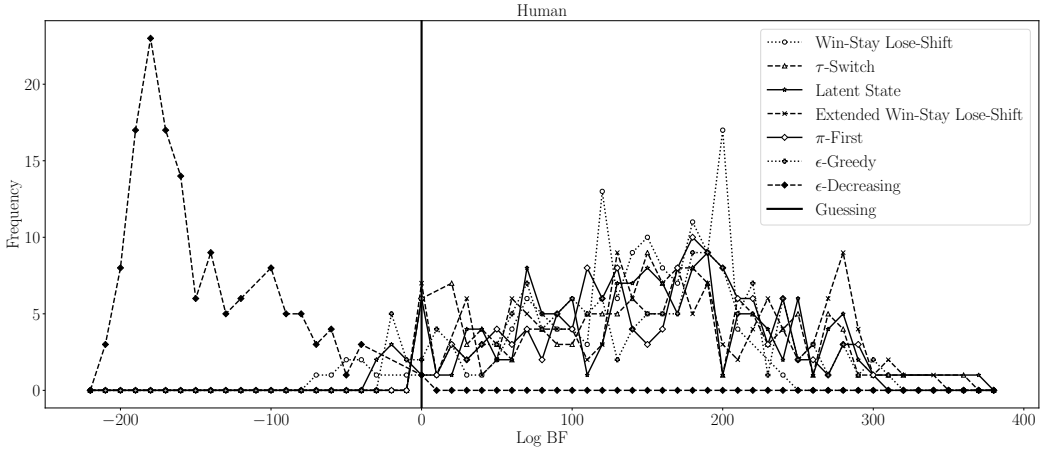
Figure 5.3: MAP Estimates on artificial data

τ-Switch and Latent state models on this particular set of bandit problems, I must take caution when making any conclusions regarding these models.

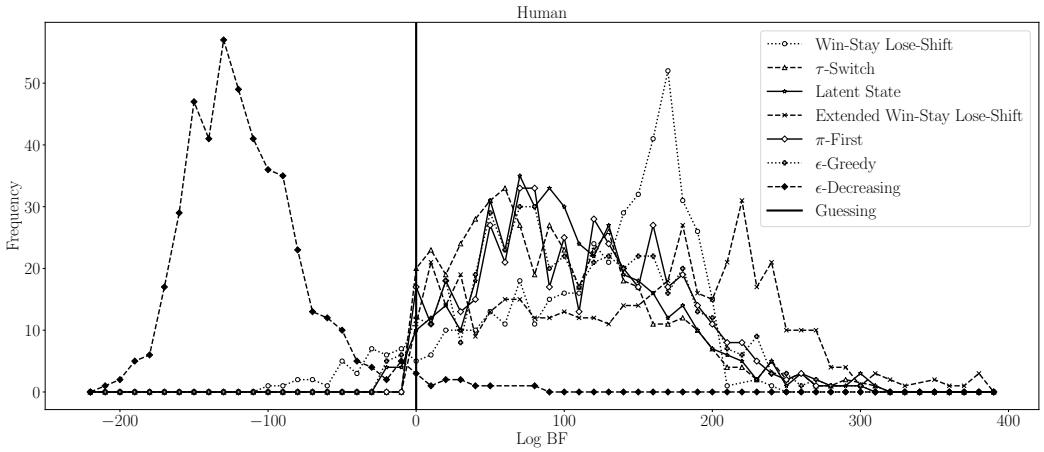
5.1.2 Individual Differences

After completing the model identifiability study, I turned my attention to the application of the models to the OED dataset. Figure 5.4a shows the log BF values for each model on the experimental data. Figure 5.5a shows the distribution of participants to models, based on which model had the highest log BF value on the human-generated dataset.

As discussed already, these results are not presented in order to determine the true models used among experimental participants. They are presented here in order to

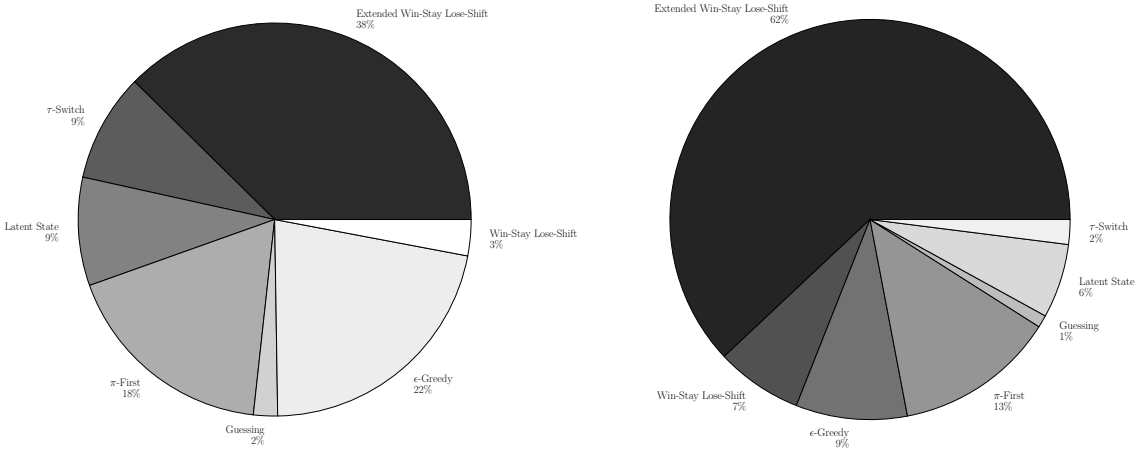


(a) Results for OED



(b) Results for testweek data

Figure 5.4: Log BF measures on experimental data across 2 datasets



(a) Results for OED

(b) Results for testweek data

Figure 5.5: Proportion of participants using each model according to maximum value of log BF measure across 2 datasets

show the relationship between models on this particular dataset. In addition to this, the analysis allows for an investigation into the informativeness of an experimental design. As such, the results of this year's analysis are presented alongside the results of a similar analysis conducted last year on the testweek dataset so as to compare the effectiveness of the design produced under the DO framework against a naïve design. To allow for a fair comparison, I reconducted the analysis from last year on the testweek dataset using the corrected τ-Switch and Latent State models, as well as the correction to the ϵ-Decreasing model[5]. To keep comparisons consistent, I also excluded results from the Optimal model from testweek data, since this was unavailable for the OED dataset.

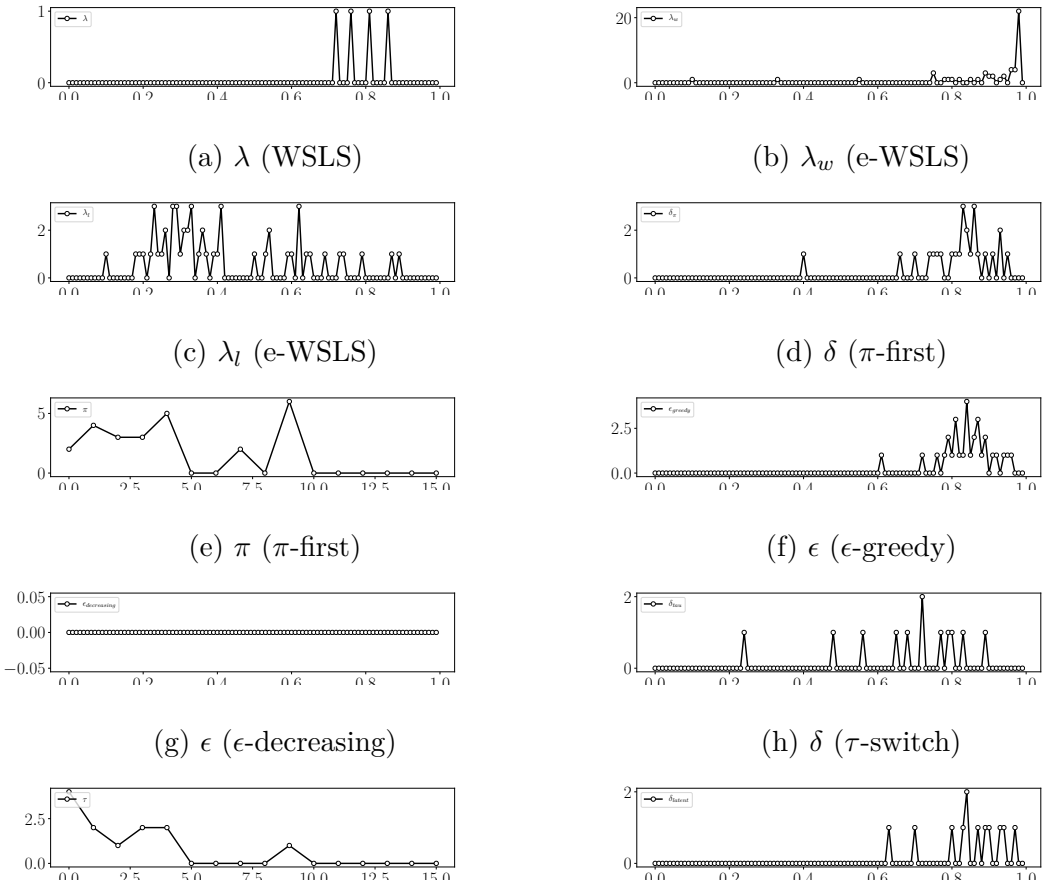
Figure 5.4a shows a large variation in log BF values across all models; this is no differ-

[5]Complete results of the model identifiability study on the testweek data set using these corrected models, alongside the original results from last year's analysis, are shown in the Appendix to this report, so as not to distract from this year's analysis.

ent from the results of the same analysis conducted on the testweek dataset showin in Figure 5.4b. The differences in the datasets become evident when looking at Figure 5.5. Whereas in the testweek dataset, 62% of participants were believed to subscribe to the e-WSLS model, this number has dropped to 38% in the OED dataset. The percentage of participants using the WSLS model has also dropped, from 7% to 3%. We see an increase in participants using the τ-Switch and Latent State models, an increase from 2% to 9%, and 7% to 9% for each model, respectively. I should note here, however, that the model identifiability and parameter recovery results for those models were dubious, so these results should be examined with caution. We see a similar increase in π-First use: up from 13% to 18%. The number of participants believed to be using the Guessing model is roughly the same across both datasets.

Turning now to parameter values, Figure 5.6 shows the MAP estimates of parameter values conditional on the best model. The mean and standard deviation of the parameter estimates calculated conditional on that model giving the best fit, as well as unconditionally across all participants, are also summarised in Table 5.2. We can see a difference in the table between unconditional and conditional MAP estimates, indicating that players who are best described by a particular model play very differently from those not best described by that model. For participants best described by the e-WSLS model, λ_w is slightly higher than when averaged across all participants, and λ_l drops from 0.552 to 0.413. In general, the accuracy of execution parameters are higher when calculated conditional on a particular model best describing a dataset.

The goal of the DO framework was to produce a design which induced participants to use a decision-making model other than the e-WSLS model, as well as producing a design which maximally differentiates between e-WSLS and all other models. I



(a) λ (WSLS)

(b) λ_w (e-WSLS)

(c) λ_l (e-WSLS)

(d) δ (π-first)

(e) π (π-first)

(f) ϵ (ϵ-greedy)

(g) ϵ (ϵ-decreasing)

(h) δ (τ-switch)

(i) τ (τ-switch)

(j) δ (Latent State)

Figure 5.6: MAP Estimates on human data, conditional on that model giving the best support for that data.

believe that the model recovery study has provided evidence that the latter goal was achieved: while e-WSLS and WSLS models may have been indistinguishable, the degree to which they were differentiated from other models was more significant than with naïve designs, as shown in Figure 5.2. There is also some support in favor of the claim that the design succeeded in influencing participants away from using the e-WSLS model: Figure 5.5 shows a decrease from 62% to 38% in the proportion of participants using the e-WSLS model.

The purpose of producing a design as described above was to falsify the theory that the majority of people use the e-WSLS model when solving bandit problems. I do

Parameter	Unconditional	Conditional
$\epsilon_{\epsilon-Greedy}$	0.729 (0.165)	0.84 (0.07)
$\epsilon_{\epsilon-Decreasing}$	0.989 (0.017)	- (-)
λ_{WSLS}	0.674 (0.097)	0.787 (0.053)
$\lambda_{w,e-WSLS}$	0.829 (0.19)	0.895 (0.163)
$\lambda_{l,e-WSLS}$	0.552 (0.199)	0.413 (0.191)
$\pi_{\pi-First}$	1.609 (3.444)	4.28 (3.156)
$\lambda_{\pi-First}$	0.712 (0.179)	0.812 (0.109)
$\lambda_{LatentState}$	0.692 (0.171)	0.845 (0.094)
$\gamma_{\tau-Switch}$	0.708 (0.174)	0.677 (0.172)
$\tau_{\tau-Switch}$	1.725 (3.698)	2.25 (2.521)

Table 5.2: MAP estimates of parameters on human data, conditional on the best model, and not

not believe that this claim has been falsified yet. While the proportion of participants using the e-WSLS model have dropped when compared to a naïve design, the e-WSLS model still has the most substantial proportion of participants of all models, as shown in Figure 5.5. Of course, this analysis shows only the relation between models on this dataset using the log BF measure for model selection. The results of the model identifiability study also cast doubt on the accuracy of models with this particular dataset and model selection method. These combined factors indicate that we cannot make claims to the ground truth of model use based on the information above; to do so, we should turn to the analysis of Bayesian models, which we will do now.

5.2 Hierarchical Models

To conduct a Bayesian analysis of the experimental data, I was able to re-use code from the first year of this project, with only minor changes made to account for the different number of arms, trials, and problems that people were faced with in the OED experiment. The code involved the implementation of each of the graphical models detailed in Chapter 3. These models were implemented in JAGS (Plummer et al., 2003), with post-processing of the Markov chain Monte Carlo (MCMC) samples conducted in R[6] using the *rjags*[7] package. Each model was fit to the OED data and 2000 samples from 2 chains were collected, with the first 1000 discarded as burn-in samples.

5.2.1 Characterization of Human Decision-Making

One goal set out in the first year of this project was to test the agreement of a set of newly proposed mixture models with data collected from human experiments with bandit problems. Of all models considered, the per-trial e-MM gave the best account of human performance on bandit problems captured in the testweek dataset (Laverty, 2019). In the conclusion of last year's report, I mentioned that while this result shows the promise the extended mixture model to account for human data, it does so on only one dataset. For a model to be deemed useful, it must be tested on new datasets to see whether this result holds. I am now in the position to test this model on new data.

To answer the question of which model gives the best account of human data, we

[6]https://www.r-project.org/

[7]https://cran.r-project.org/web/packages/rjags/index.html

Model	Minimum	Maximum	Mean	Standard Dev.
ϵ-Greedy	0.174	0.785	0.567	0.137
ϵ-Decreasing	0.181	0.767	0.559	0.137
WSLS	0.151	0.746	0.517	0.128
Extended-WSLS	0.192	0.88	0.571	0.170
π-First	0.187	0.759	0.544	0.130
Latent State	0.177	0.834	0.6	0.149
τ-Switch	0.173	0.805	0.578	0.138
Mixture Model (per participant)	0.166	0.882	0.628	0.153
Mixture Model (per game)	0.231	0.89	0.65	0.144
Mixture Model (per trial)	0.201	0.884	0.666	0.151
Extended Mixture Model (per participant)	0.192	0.884	0.631	0.148
Extended Mixture Model (per game)	0.221	0.89	0.665	0.141
Extended Mixture Model (per trial)	**0.207**	**0.876**	**0.672**	**0.150**

Table 5.3: Summary of test statistics from posterior predictive agreement analysis

can calculate the posterior predictive agreement (PPA) of a model with a dataset. PPA is used in statistics and the cognitive sciences to assess the descriptive ability of a Bayesian model (Gelman et al., 2013; Shiffrin et al., 2008). The posterior predictive of a model is calculated by taking the predictions made by a model at all possible parameter settings and weighing those according to the posterior probability of each setting. Since the posterior predictive takes into account predictions made at all possible parameter settings and averages results across all of these settings, it automatically balances goodness-of-fit with model complexity. This property is essential

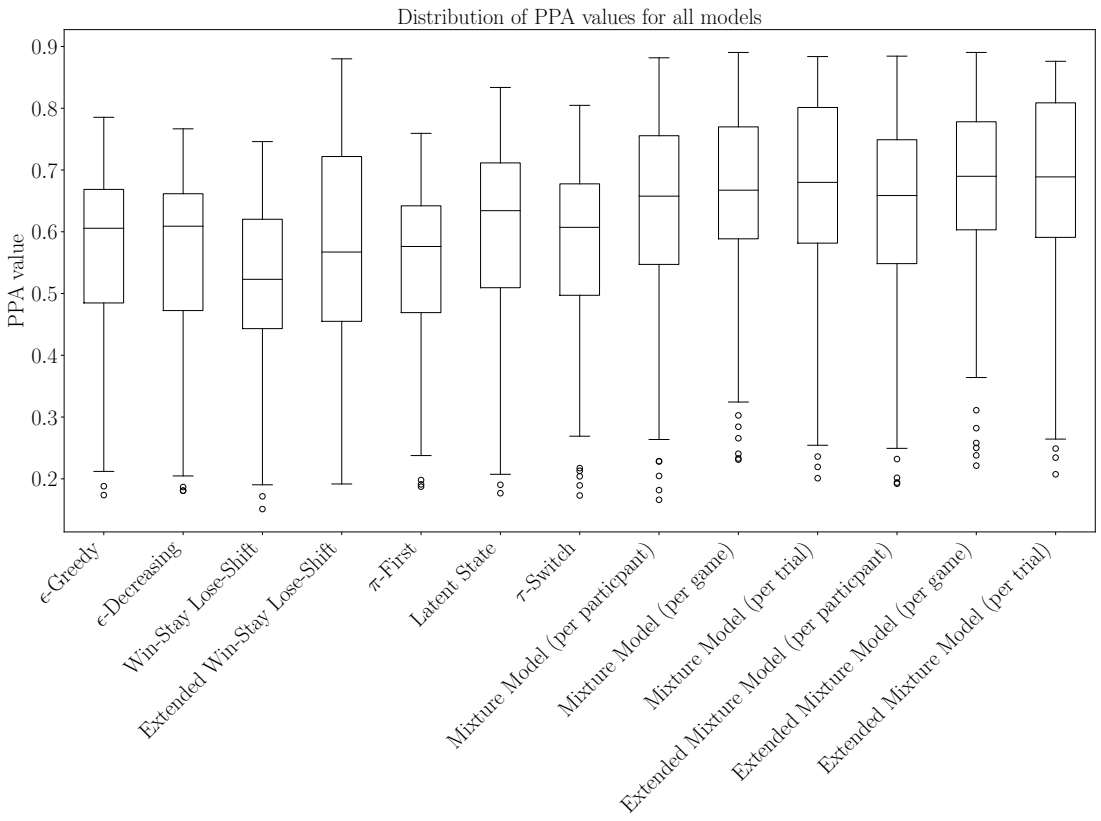


Figure 5.7: Visualisation of results from posterior predictive agreement analysis

in comparing simpler models like the ϵ-Greedy model with complex mixture models. The PPA of each model with the OED was calculated, and a summary of those values are shown in Table 5.3; a box-plot visualization showing the median PPA value, along with maximum, minimum, and interquartile range for the 138 participants in the OED dataset is shown in Figure 5.7.

While Table 5.3 shows a summary of posterior predictive values across all participants, Figure 5.8 shows the distribution of participants to models based on which model had the highest PPA value on that participant's data. We see that not only did the per-trial e-MM have the largest average PPA value, it also had the highest PPA value for most participants: the model best described 39.1% of participants. In fact, 86.3% of participants were best described by some form of mixture model. These results are incredibly similar to the results of a PPA analysis conducted on

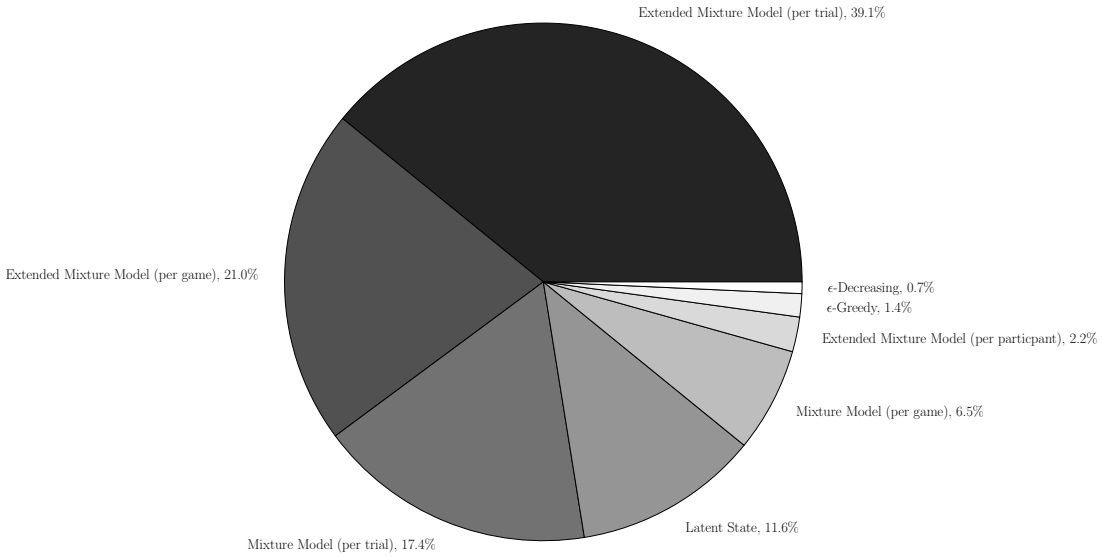


Figure 5.8: Percentage of participants whose decisions had largest posterior predictive agreement with a given model, OED data

the testweek dataset in last year's project, shown in Figure 5.9. The most significant differences in the two sets of results are the decrease of 12.8% of participants best described by the per-trial e-MM, and the increase of 9.6% in the percentage of participants using the simpler per-trial MM. It is interesting to note that while a simpler mixture model best describes a larger proportion of participants in the OED dataset, the overall percentage of participants best described by a mixture model allowing for per-trial switching has not changed much: 56.5% in the OED dataset versus 59.7% in the testweek dataset. To further investigate this shift in model use, we can turn to a posterior predictive analysis of the parameter which describes model use in a mixture model.

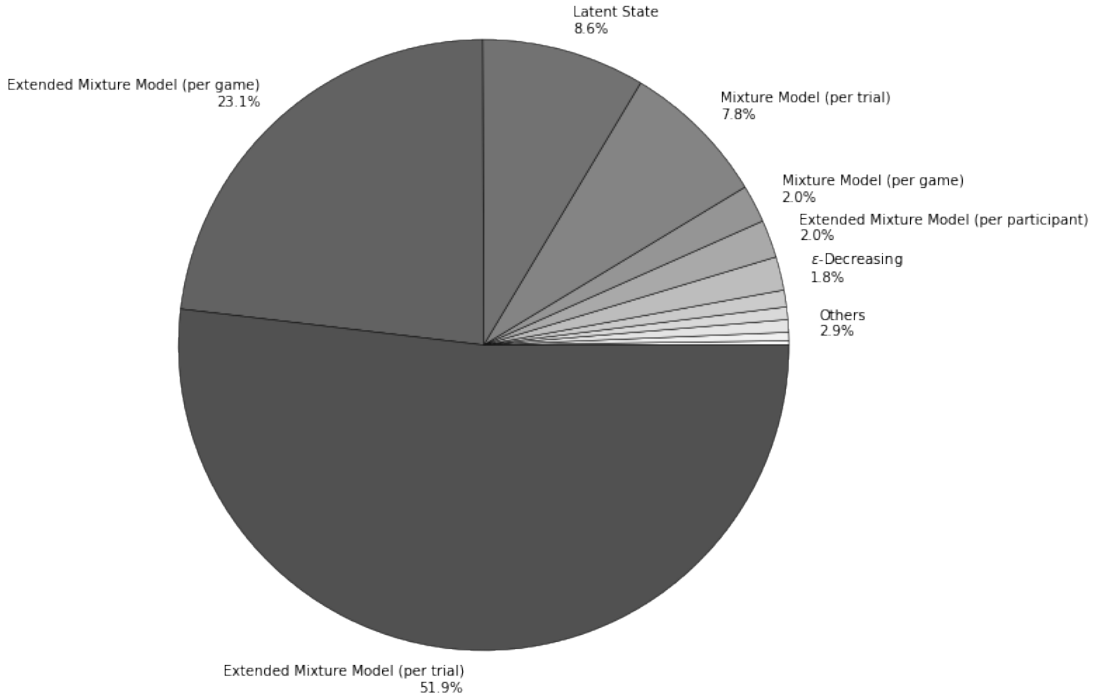


Figure 5.9: Percentage of participants whose decisions had largest posterior predictive agreement with a given model, testweek data. Reproduced from (Laverty, 2019)

5.2.2 Model Differences

As described in Chapter 3, all mixture models have an assignment parameter, z, which, along with the mixture process, h, describes which process was used to generate observed data. The assignment parameter is believed to be generated according to a categorical distribution, $z \sim Categorical(\phi)$. By examining the posterior expectation of ϕ after a model has been fit to data we can infer the proportions of participants believed to be using each component model in the mixture model. The resulting proportions for each component model are shown for each mixture model in Table 5.4. The rows of Table 5.4 show the proportion of participants using that model in each of the mixture models. The top row of each cell shows the results after the models were fit to the OED dataset, and the bottom row of each cell shows the results for

Model	Mixture Model (participant)	Mixture Model (game)	Mixture Model (trial)	Extended Mixture Model (participant)	Extended Mixture Model (game)	Extended Mixture Model (trial)
WSLS	4.65%	5.65%	0.01%	2.47%	2.02%	0.01%
	2.72%	5.73%	0.00%	3.07%	2.80%	0.00%
e-WSLS	41.0%	**43.7%**	**54.5%**	**40.1%**	**35.1%**	**53.1%**
	68.2%	**63.7%**	**72.2%**	**65.1%**	**60.7%**	**72.0%**
ϵ-Greedy	**53.1%**	28.3%	45.4%	32.5%	5.57%	43.1%
	27.7%	11.6%	26.2%	8.57%	0.460%	11.6%
ϵ-Decreasing	1.26%	22.4%	0.04%	0.24%	16.2%	0.05%
	1.32%	19.0%	1.54%	0.57%	9.45%	10.8%
π-First	-	-	-	22.2%	27.0%	0.01%
	-	-	-	17.2%	19.4%	0.00%
Latent State	-	-	-	1.94%	8.38%	3.67%
	-	-	-	0.35%	0.02%	0.00%
τ-Switch	-	-	-	0.56%	5.71%	0.01%
	-	-	-	5.10%	7.11%	5.59%

Table 5.4: Proportion of participants using each model in all mixture models in OED data (top row) and testweek data (bottom row)

the testweek dataset for comparison[8]. Reading the columns of Table 5.4 shows the distribution of component model use within each mixture model, with the model with

[8]testweek dataset results are reproduced from (Laverty, 2019, Table 5.4)

the largest support highlighted in bold.

To investigate the shift from the per-trial e-EMM users to the simpler per-trial MM seen at the end of the preceding section, we can look at the corresponding columns of Table 5.4. We see that for the per-trial e-MM, the majority of participants - 96.2% - use either the e-WSLS or ϵ-Greedy models, with only 3.69% of participants using a model not included in the per-trial MM. Contrast this result to the testweek dataset where 5.59% of participants used a component model which was not available in the simpler per-trial MM. Looking at exact numbers, around 25 testweek participants, compared to 5 OED participants, used a component model only available in the e-MM. It is possible to explain this shift in model use, therefore, as a preference of the simpler model over the more complex variant when there was a much smaller number of participants benefiting from the additional complexity of the extended mixture model.

Once again, the primary goal of this project was to test the claim that the majority of humans use the e-WSLS decision-making strategy when solving bandit problems. We are now in a position, by analyzing the posterior estimates of the mixture parameter of the mixture models, to determine the proportion of participants in the OED dataset inferred to be using the e-WSLS model. The values highlighted in bold in each column of Table 5.4 identify the component model which was used by the majority of participants in each mixture model. We can see that besides one mixture model, the e-WSLS model was identified as being the dominant model used when solving the OED bandit problems. An interesting observation when comparing the results of the e-WSLS model in the OED and testweek datasets is that the proportion of participants using the e-WSLS model dropped in all cases, dropping by an average of 22.4%.

This result indicates a success in the DO framework to induce participants to use a model other than e-WSL; it seems that the majority of participants drifted towards the ϵ-Greedy model, with a small proportion using the Latent State model. When designing a game using the DO framework, I sacrificed some utility in favor of variability. It would be interesting to conduct additional experiments which favor higher utility to test whether participants are further driven to use alternate models, and whether any proportion of alternate model use is high enough to knock the e-WSLS model off of its pedestal. However, we have evidence here that the DO framework succeeded in inducing participants away from using the e-WSLS model, but that the majority of participants stuck with it in anyhow, giving credence to the theory that the majority of people use the e-WSLS model when solving bandit problems.

5.2.3 Parameter Differences

Of final interest is the difference in parameter use across participants and models in the OED dataset. A primary reason for the use of cognitive models in the study of bandit problems is that the parameters of these models have psychologically interpretable meaning; analyzing the posterior estimates of these parameters after applying a model to a dataset of human decisions on bandit problems allows researchers to glean insights into how people deal with uncertainty, and explore the balance between exploration and exploitation when confronted with an unknown environment.

Table 5.5 shows the posterior estimates of the key parameters of all models considered in this project. We can see that considering a cognitive model individually versus as part of a mixture model has a substantial effect on the posterior estimates for that model's parameters. This difference in values is due to the fact that mixture

Parameter	Individual	ZLMM	MyMM	ZLMM Full	MyMM Full	ZLMM Trial	MyMM Trial
$\epsilon_{\epsilon-Greedy}$	0.294 (0.159)	0.253 (0.140)	0.164 (0.0683)	0.281 (0.160)	0.21 (0.0988)	0.129 (0.181)	0.132 (0.190)
$\epsilon_{\epsilon-Decreasing}$	0.982 (0.0252)	0.729 (0.1303)	0.56 (0.1466)	0.41 (0.2386)	0.159 (0.0864)	0.472 (0.1369)	0.293 (0.1555)
λ_{WSLS}	0.673 (0.0944)	0.707 (0.1151)	0.699 (0.118)	0.743 (0.1343)	0.783 (0.1229)	0.486 (0.1549)	0.507 (0.1516)
$\lambda_{w,e-WSLS}$	0.833 (0.184)	0.923 (0.0608)	0.924 (0.0708)	0.809 (0.2601)	0.948 (0.0626)	0.824 (0.2213)	0.829 (0.2209)
$\lambda_{l,e-WSLS}$	0.553 (0.194)	0.391 (0.1481)	0.382 (0.1347)	0.456 (0.2994)	0.32 (0.1961)	0.474 (0.3238)	0.465 (0.3297)
$\pi_{\pi-First}$	8.04 (6.52)	-	10.458 (5.7883)	-	10.294 (5.6864)	-	10.505 (5.7683)
$\lambda_{\pi-First}$	0.156 (0.189)	-	0.235 (0.1656)	-	0.158 (0.1933)	-	0.252 (0.1943)
$\lambda_{LatentState}$	0.724 (0.174)	-	0.744 (0.1207)	-	0.889 (0.078)	-	0.9 (0.0629)
$\gamma_{\tau-Switch}$	0.696 (0.160)	-	0.517 (0.1469)	-	0.774 (0.1145)	-	0.46 (0.168)
$\tau_{\tau-Switch}$	13.6 (6.66)	-	10.505 (5.7719)	-	10.485 (5.7668)	-	10.5 (5.7668)

Table 5.5: Means and standard deviations of posterior estimates for key psychological parameters in each cognitive model

models calculate posterior estimates of a component model's parameters conditional on that model giving the best account of the observed data. Only data points inferred to be generated by a model are used to update the posterior estimates for that model's parameters (M. D. Lee, 2018). By considering this, we can compare the average posterior estimates for all participants - the *Individual* column in Table 5.5 - against the posterior estimates for only those participants inferred to be using that model, i.e., the remaining columns of Table 5.5. While I won't go into detail on how each parameter differs according to each mixture modeling strategy used, I believe the observed difference in posterior parameter estimates highlights the benefit in using hierarchical mixture models for capturing individual differences in a dataset.

5.2.4 Characterization of Optimal Decision-Making

> *"Three basic challenges in studying any real-world decision-making problem are to characterize how people solve the problem, characterize the optimal approach to solving the problem, and then characterize the relationship between the human and optimal approach."* - (M. D. Lee et al., 2011)

The final stages of this analysis will be turned towards a brief analysis of JAGS models on optimal decision data for the OED. The benefit of such an analysis is that we can compare the posterior estimates of parameters calculated on optimal decision data with those estimates calculated on human data. Such a comparison allows us to view where human performance fell short of optimal, allowing for an opportunity to instruct humans on how they should solve the problem. In addition to this, testing the ability of a model to describe optimal data allows us to determine whether a heuristic model could be used as a substitute for the computationally expensive optimal model.

To generate the optimal data, the Scala code detailed in section 4.2.1 was used, with the Eddie compute cluster again used to manage the computational complexity of this method. Only 11 samples were able to be generated, as any more would have taken too long given the time frame of the project.

Model	Minimum	Maximum	Mean	Standard Dev.
ϵ-Greedy	0.22	0.56	0.36	0.09
ϵ-Decreasing	0.25	0.57	0.38	0.08
Win-Stay Lose-Shift	0.47	0.57	0.52	0.03
Extended Win-Stay Lose-Shift	0.51	0.63	0.58	0.04
π-First	0.32	0.52	0.4	0.06
Latent State	0.39	0.56	0.48	0.06
τ-Switch	0.38	0.54	0.46	0.05
Mixture Model (per participant)	0.5	0.63	0.58	0.04
Mixture Model (per game)	0.53	0.64	0.59	0.04
Mixture Model (per trial)	0.55	0.68	0.61	0.04
Extended Mixture Model (per participant)	0.51	0.64	0.58	0.04
Extended Mixture Model (per game)	0.55	0.65	0.6	0.03
Extended Mixture Model (per trial)	**0.57**	**0.7**	**0.64**	**0.04**

Table 5.6: Summary of test statistics from posterior predictive agreement analysis on optimal data

The first result worth reporting is the ability of each model to give a good account of the optimal data; the posterior predictive agreement of each model is shown in Table 5.6, with a box-plot of the data shown in Figure 5.10. We see in Table 5.6 that, similar to the human data, the per-trial e-MM had the highest mean PPA on the given dataset. An analysis of the PPA value of each model on each optimal data sample showed the per-trial e-MM had the highest PPA value in all 11 samples. The average values are not all too high for any model, indicating that no model is an ideal candidate to substitute for the optimal model on this set of bandit problems. Among the models which best accounted for both optimal and human data were mixture models which allowed for switching between model use on a per-game and per-trial

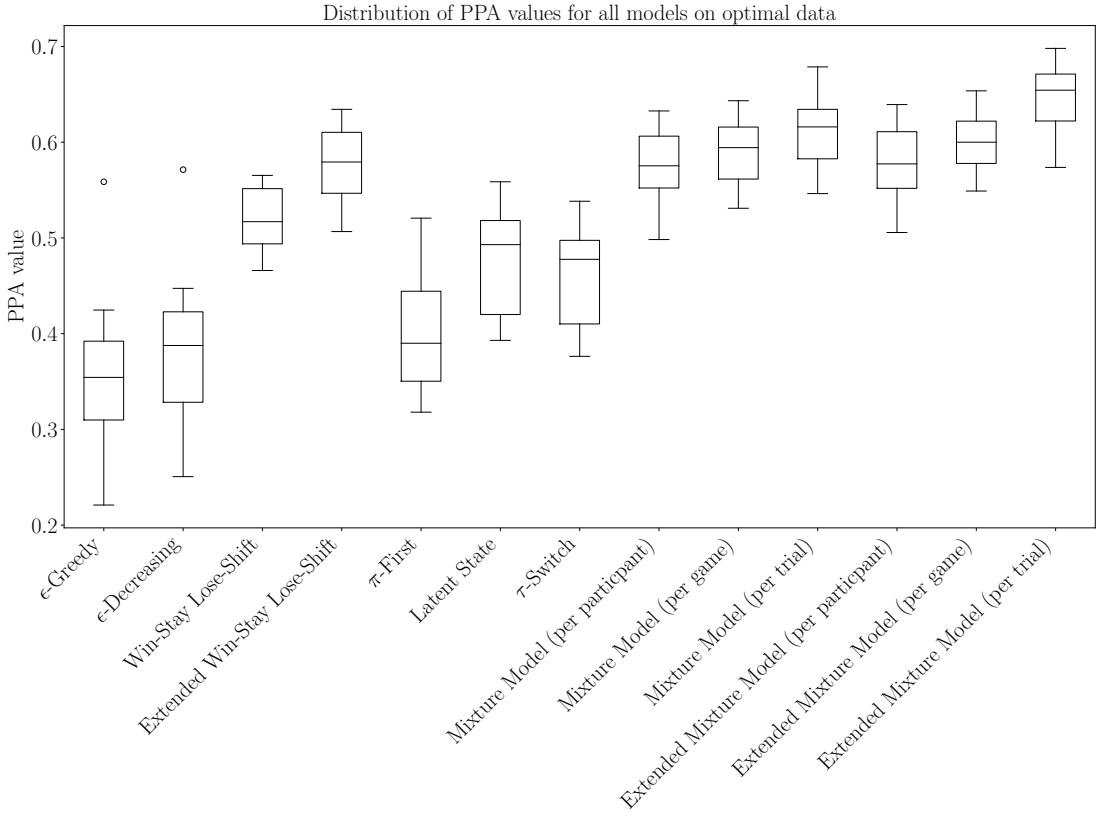


Figure 5.10: Boxplot of results from posterior predictive agreement analysis on optimal data

basis, providing more evidence that these flexible latent state models are of value in analyses of human decision making on bandit problems.

Parameter	Human Data	Optimal Data
$\epsilon_{\epsilon-Greedy}$	0.132 (0.190)	0.518 (0.148)
$\epsilon_{\epsilon-Decreasing}$	0.293 (0.155)	0.494 (0.169)
λ_{WSLS}	0.507 (0.152)	0.464 (0.146)
$\lambda_{w,e-WSLS}$	0.829 (0.221)	0.961 (0.040)
$\lambda_{l,e-WSLS}$	0.465 (0.330)	0.288 (0.074)
$\pi_{\pi-First}$	10.5 (5.77)	11.5 (6.23)
$\lambda_{\pi-First}$	0.252 (0.194)	0.058 (0.062)
$\lambda_{LatentState}$	0.900 (0.063)	0.731 (0.140)
$\gamma_{\tau-Switch}$	0.460 (0.168)	0.476 (0.151)
$\tau_{\tau-Switch}$	10.5 (5.77)	10.5 (5.77)

Table 5.7: Posterior estimates of component model parameters in the e-MM

The question of how humans compare to the optimal model can be answered by looking at the results of Table 5.7. The posterior estimates for human data are close to optimal in many cases: where the accuracy of execution parameters from the optimal data are high, e.g., $\lambda_{w,e-WSLS}$, and $\lambda_{LatentState}$, human values are high as well, and when they are low, e.g., $\gamma_{\tau-Switch}$ and $\lambda_{l,e-WSLS}$, those values are also lower in the human data. However, the exact values of the human parameters are slightly off of optimal: $\lambda_{w,e-WSLS}$ is not as high as the optimal data suggests it should be, nor is the value of $\lambda_{l,e-WSLS}$ as low as the optimal data dictates. There are certain cases where

the estimated human values are from their optimal counterparts: see the results for $\epsilon_{\epsilon-Greedy}$, $\epsilon_{\epsilon-Decreasing}$, and $\lambda_{\pi-First}$. The gap between human and optimal parameter estimates indicates room for human improvement, highlighting the value of comparing the two sets of estimates. Similar results were reported in the first year of this project when a similar analysis was conducted on the testweek dataset (Laverty, 2019). As was suggested in the concluding remarks of last year's analysis, it would be interesting to use these optimal parameter estimates in an experiment which instructed participants on how they should behave in order to solve the problem optimally, and to see whether such an intervention improves participant performance when solving future problems.

Chapter 6

Conclusion

6.1 Contributions of Work

The main goal of this project was to test the theory that the e-WSLS model is the dominant strategy people use when solving bandit problems. In order to test this theory, I attempted to falsify it.

In order to do so, I first had to develop a DO framework that would produce an OED which could be used in a confirmatory experiment to test such a theory. This framework was developed to produce a design where the e-WSLS model performed poorly relative to other models, where it was maximally differentiated from those other strategies. In doing so, I believed that the design produced would induce participants in an experiment to use a model other than the e-WSLS model, thus falsifying the claim that it is a dominant decision-making strategy. The criteria that only one model need be differentiated from all others, and that the chosen model should perform poorly relative to others are unique to this project. The DO framework developed here is the first framework proposed in the DO literature on bandit problems to

use such criteria, with previous attempts focusing solely on model discrimination. I believe that these criteria could be used by future researchers to design experiments that rigorously test theories related to a single model.

I proposed two separate methods for finding designs where a model performs poorly: the first is to directly optimize for this criterion by including model performance on a design in a local utility function; the second is to pre-filter designs based on model performance prior to design optimization. Where the first method is highly automated, the second allows for greater flexibility and control in experimental design. The benefit of the latter approach was highlighted in discussions regarding the utility-variability tradeoff which arose during experimental design. While not used in this project, I believe that the former still holds value, and could be of particular use to future researchers, especially those using statistical sampling approaches, as opposed to the grid search I conducted in this project.

When developing the framework, I also considered a broader range of designs than ever done before on bandit problems, optimizing: the number of trials in a problem, the number of arms in a problem, the distribution of rewards across arms, and the division of an experiment with a fixed number of trials into multiple problems. In analyzing designs produced by this framework, I concluded the effect each of these factors has on the utility of a design, allowing me to answer previously posed questions such as that of the tradeoff between the number of problems, and the length of those problems, in experiments involving bandit problems Zhang and Lee (2010b). The number of design factors considered in this project is the largest of any attempt to apply DO techniques to bandit problems.

I applied this framework to the task of producing an OED which could falsify the

theory outlined above. In doing so, I provided evidence in Chapter 4 that the DO framework succeeded in identifying designs where e-WSLS was both maximally differentiated from other models, and performed poorly relative to those models. I used an OED produced by the framework to conduct an experiment with 138 participants, making this the first study to use an OED in a real experiment testing human performance on bandit problems. The dataset produced from this experiment is, therefore, the first of it's kind, and will surely be of use to future researchers studying the problem.

In analyzing this dataset using hierarchical Bayesian models, I have provided evidence that suggests that while fewer people use the e-WSLS model in the OED when compared to a naïve design - indicating a success of the OED to induce alternate model use - a posterior analysis of the mixture parameter in the best fitting mixture model showed that the majority of participants were inferred to be using the e-WSLS model. The falsification of the theory outlined above was, therefore, unsuccessful, thus giving credence to the original theory. This result suggests that the e-WSLS is perhaps an example of a fast-and-frugal heuristic that people use when making decisions (Todd & Gigerenzer, 2000).

I was also able to analyze the performance of extended mixture models I developed last year on this new dataset via a posterior predictive agreement analysis. The results of this analysis provide more evidence that these more flexible and robust models gave the best account of human data than all other models considered.

All of the code developed throughout this project is highly flexible, allowing users to change the conditions for generating and filtering, bandit designs via simple command-line arguments. The code is also not specific to the eWSLS model: the

model for which utilities are calculated can also be changed via command-line arguments so that OEDs for other models can be developed. I hope that future researchers interested in experimental design for bandit problems could use it to generate bandits under optimal conditions to test their specific theories. Documentation is provided on how to configure the code, and it will be available to any interested researchers.

6.2 Future work

The DO framework developed in this project was based on the framework developed by Myung and Pitt (2009). Some researchers have pointed out limitations in simple DO, and have proposed the use of Adaptive Design Optimization (ADO) frameworks instead (Sun, 2012). ADO is a sequential extension to DO, adapting the design based on the results gathered from participant's decisions thus far. The benefit of this approach is that designs can be tailored to the participants of the study, allowing for more robust conclusions to be drawn. The downside is that calculating the utility of designs is a costly process, and this must be done in an online manner as data is gathered. Therefore, the complexity of the designs considered in an ADO are somewhat limited, and experiments are much longer than those where designs are produced by the simpler DO prior to conducting the experiment. ADO is very successful in discriminating between a small number of models (Sun, 2012), and so I believe that using it in an experiment comparing only, say, the e-WSLS and ϵ-Greedy models - the models with the largest support in the OED dataset - could be possible, and very worthwhile in gathering more evidence in favor/against the theory proposed in this project.

One criterion in my DO framework was that it should produce a design in which people should not use the e-WSLS model. This was quantified by producing a design where the e-WSLS model performed poorly relative to other models. An alternate approach could have been to use the posterior estimates of the mixture parameter of a mixture model which was fit to simulated data generated for a specific design. The design in which the e-WSLS model had the lowest support according to this parameter could then be selected. However, as both the grid search and fitting of JAGS models took a long time, this approach was not possible given the time constraints of the project. I believe, however, that this approach could be made possible in future projects with fewer time constraints by using either sampling techniques, or a distributed computing approach as was used in this project.

Finally, this project used only one type of design in an experiment, a design discriminating against the e-WSLS model with results compared to a previous experiment using a naïve design. An alternate experiment could use multiple designs: a naïve design not subject to any DO techniques; a design optimized to discriminate against the e-WSLS model - as is done in this project - and a design optimized in favor of the e-WSLS model. The use of a control group alongside positive and negative effect groups would allow us to better test the effect of DO techniques on model use, and allow for more robust conclusions to be drawn.

References

Aston-Jones, G., & Cohen, J. D. (2005). An integrative theory of locus coeruleus-norepinephrine function: adaptive gain and optimal performance. *Annu. Rev. Neurosci.*, *28*, 403–450.

Berry, D. A., & Fristedt, B. (1985). Bandit problems: sequential allocation of experiments (monographs on statistics and applied probability). *London: Chapman and Hall*, *5*, 71–87.

Burtini, G., Loeppky, J., & Lawrence, R. (2015). A survey of online experiment design with the stochastic multi-armed bandit. *arXiv preprint arXiv:1510.00757*.

Cavagnaro, D. R., Myung, J. I., Pitt, M. A., & Kujala, J. V. (2010). Adaptive design optimization: A mutual information-based approach to model discrimination in cognitive science. *Neural computation*, *22*(4), 887–905.

Christian, B., & Griffiths, T. (2016). *Algorithms to live by: The computer science of human decisions*. Macmillan.

Ecdf. (n.d.). Retrieved 2020-04-12, from `https://www.ecdf.ed.ac.uk` (University of Edinburgh)

El-Gamal, M. A., & Palfrey, T. R. (1996). Economical experiments: Bayesian efficient experimental design. *International Journal of Game Theory*, *25*(4), 495–517.

Farrell, S., & Lewandowsky, S. (2018). *Computational modeling of cognition and*

behavior. Cambridge University Press.

Fisher, R. A. (1936). Design of experiments. *Br Med J*, *1*(3923), 554–554.

Gelman, A., Stern, H. S., Carlin, J. B., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian data analysis*. Chapman and Hall/CRC.

Green, D. M., & Luce, R. D. (1971). *Speed-accuracy tradeoff in auditory detection*.

Guan, M., Stokes, R., Vandekerckhove, J., & Lee, M. D. (n.d.). A cognitive modeling analysis of risk in sequential choice tasks.

Gutierrez, M., Cernỳ, J., Ben-Asher, N., Aharonov, E., Basak, A., Bošanskỳ, B., ... Gonzalez, C. (n.d.). Evaluating models of human behavior in an adversarial multi-armed bandit problem.

Heathcote, A., Brown, S., & Mewhort, D. (2000). The power law repealed: The case for an exponential law of practice. *Psychonomic bulletin & review*, *7*(2), 185–207.

Heck, D. W., & Erdfelder, E. (2019). Maximizing the expected information gain of cognitive modeling via design optimization. *Computational Brain & Behavior*, *2*(3-4), 202–209.

Hills, T. T., Todd, P. M., Lazer, D., Redish, A. D., Couzin, I. D., Group, C. S. R., et al. (2015). Exploration versus exploitation in space, mind, and society. *Trends in cognitive sciences*, *19*(1), 46–54.

Hunt, E., Frost, N., & Lunneborg, C. (1973). Individual differences in cognition: A new approach to intelligence. In *Psychology of learning and motivation* (Vol. 7, pp. 87–122). Elsevier.

Ivan, V. E., Banks, P. J., Goodfellow, K., & Gruber, A. J. (2018). Lose-shift responding in humans is promoted by increased cognitive load. *Frontiers in integrative*

neuroscience, *12*, 9.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, *4*, 237–285.

Kass, R. E., & Raftery, A. E. (1995). Bayes factors. *Journal of the american statistical association*, *90*(430), 773–795.

Laverty, P. (2019). Investigating individual differences in human decision-making.

Lee, D., Conroy, M. L., McGreevy, B. P., & Barraclough, D. J. (2004). Reinforcement learning and decision making in monkeys during a competitive game. *Cognitive brain research*, *22*(1), 45–58.

Lee, M. D. (2008). Three case studies in the bayesian analysis of cognitive models. *Psychonomic Bulletin & Review*, *15*(1), 1–15.

Lee, M. D. (2011a). How cognitive modeling can benefit from hierarchical bayesian models. *Journal of Mathematical Psychology*, *55*(1), 1–7.

Lee, M. D. (2011b). Special issue on hierarchical bayesian models. *Journal of Mathematical Psychology*, *55*, 1–118.

Lee, M. D. (2018). Bayesian methods in cognitive modeling. *Stevens' Handbook of Experimental Psychology and Cognitive Neuroscience*, *5*, 1–48.

Lee, M. D., Gluck, K. A., & Walsh, M. M. (2019). Understanding the complexity of simple decisions: Modeling multiple behaviors and switching strategies. *Decision*.

Lee, M. D., Zhang, S., Munro, M., & Steyvers, M. (2009). Using heuristic models to understand human and optimal decision-making on bandit problems. In *Proceedings of the ninth international conference on cognitive modeling—iccm2009. manchester, uk.*

Lee, M. D., Zhang, S., Munro, M., & Steyvers, M. (2011). Psychological models of human and optimal performance in bandit problems. *Cognitive Systems Research*, *12*(2), 164–174.

L Griffiths, T., Kemp, C., & B Tenenbaum, J. (2008). Bayesian models of cognition.

Luce, R. D. (1995). Four tensions concerning mathematical modeling in psychology. *Annual Review of Psychology*, *46*(1), 1–27.

Myung, J. I., & Pitt, M. A. (2009). Optimal experimental design for model discrimination. *Psychological review*, *116*(3), 499.

Nowak, M., & Sigmund, K. (1993). A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner's dilemma game. *Nature*, *364*(6432), 56–58.

Pitt, M. A., Myung, I. J., & Zhang, S. (2002). Toward a method of selecting among computational models of cognition. *Psychological review*, *109*(3), 472.

Plummer, M., et al. (2003). Jags: A program for analysis of bayesian graphical models using gibbs sampling. In *Proceedings of the 3rd international workshop on distributed statistical computing* (Vol. 124, pp. 1–10).

Popper, K. (2014). *Conjectures and refutations: The growth of scientific knowledge*. routledge.

Robbins, H. (1952). Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, *58*(5), 527–535.

Roberts, S., & Pashler, H. (2000). How persuasive is a good fit? a comment on theory testing. *Psychological review*, *107*(2), 358.

Settles, B. (2009). *Active learning literature survey* (Tech. Rep.). University of Wisconsin-Madison Department of Computer Sciences.

Shen, W., Wang, J., Jiang, Y.-G., & Zha, H. (2015). Portfolio choices with orthogonal

bandit learning. In *Twenty-fourth international joint conference on artificial intelligence.*

Shiffrin, R. M., Lee, M. D., Kim, W., & Wagenmakers, E.-J. (2008). A survey of model evaluation approaches with a tutorial on hierarchical bayesian methods. *Cognitive Science*, *32*(8), 1248–1284.

Steyvers, M., Lee, M. D., & Wagenmakers, E.-J. (2009). A bayesian analysis of human decision-making on bandit problems. *Journal of Mathematical Psychology*, *53*(3), 168–179.

Sun, Y. (2012). *Adaptive design optimization for model discrimination under model misspecification* (Unpublished doctoral dissertation). The Ohio State University.

Sutton, R. S., Barto, A. G., et al. (1998). *Introduction to reinforcement learning* (Vol. 135). MIT press Cambridge.

Tamura, K., & Masuda, N. (2015). Win-stay lose-shift strategy in formation changes in football. *EPJ Data Science*, *4*(1), 9.

Todd, P. M., & Gigerenzer, G. (2000). Précis of simple heuristics that make us smart. *Behavioral and brain sciences*, *23*(5), 727–741.

Vickery, T. J., Chun, M. M., & Lee, D. (2011). Ubiquity and specificity of reinforcement signals throughout the human brain. *Neuron*, *72*(1), 166–177.

Wandell, B. A. (1977). Speed-accuracy tradeoff in visual detection: Applications of neural counting and timing. *Vision Research*, *17*(2), 217–225.

Wang, H., & Sun, C.-T. (2011). Game reward systems: Gaming experiences and social meanings. In *Digra conference* (Vol. 114).

Worthy, D. A., Hawthorne, M. J., & Otto, A. R. (2013). Heterogeneity of strategy

use in the iowa gambling task: A comparison of win-stay/lose-shift and reinforcement learning models. *Psychonomic bulletin & review*, *20*(2), 364–371.

Zeigenfuse, M. D., & Lee, M. D. (2009). Bayesian nonparametric modeling of individual differences: A case study using decision-making on bandit problems. In *Proceedings of the 31st annual conference of the cognitive science society, austin, tx: Cognitive science society* (pp. 1412–1417).

Zhang, S., & Lee, M. (2010a). Cognitive models and the wisdom of crowds: A case study using the bandit problem. In *Proceedings of the annual meeting of the cognitive science society* (Vol. 32).

Zhang, S., & Lee, M. D. (2010b). Optimal experimental design for a class of bandit problems. *Journal of Mathematical Psychology*, *54*(6), 499–508.

Appendix A

The following appendix includes the results of the analysis carried out in Section 5.2 of (Laverty, 2019), re-conducted with the correctly implemented τ-Switch, Latent State, and ϵ-Greedy models.

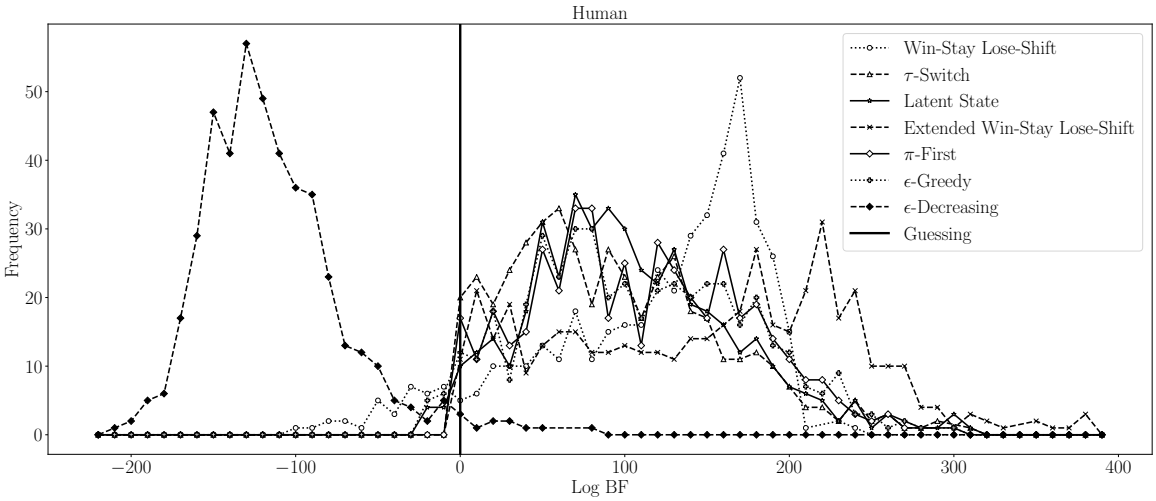
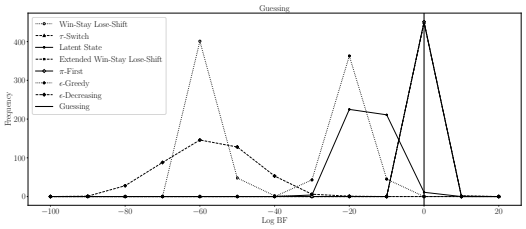


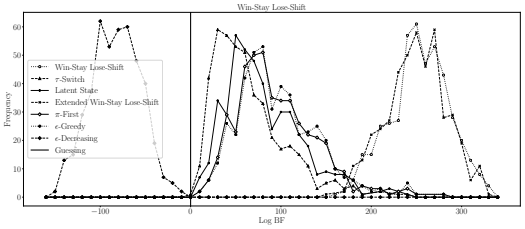
Figure A.1: Log BF measures on Human data. Compare with (Laverty, 2019, Figure 5.3).

Recovery Model	Guessing	WSLS	e-WSLS	ϵ-Greedy	ϵ-Decreasing	π-first	Latent State	τ-switch
Guessing	57	0	29	0	0	3	0	11
Win-Stay Lose-Shift	0	100	0	0	0	0	0	0
Extended Win-Stay Lose-Shift	0	100	0	0	0	0	0	0
ϵ-Greedy	0	0	0	100	0	0	0	0
ϵ-Decreasing	12	0	25	16	0	11	28	8
π-First	0	0	0	0	0	100	0	0
Latent State	0	18	45	5	0	0	31	1
τ-Switch	0	0	33	3	0	0	37	27

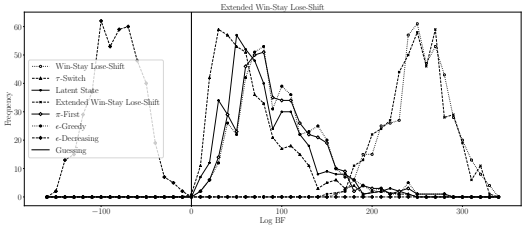
Table A.1: Model recovery on artificial data (rows indicate the model used to generate the dataset. Compare with (Laverty, 2019, Table 5.1)



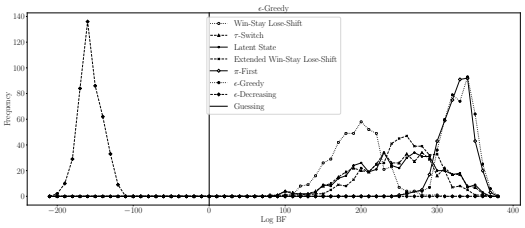
(a) Guessing



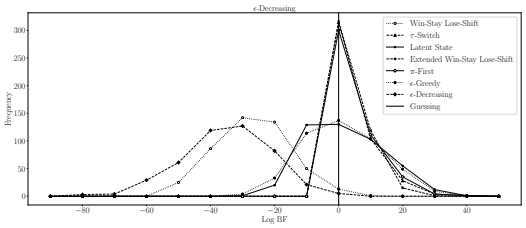
(b) WSLS



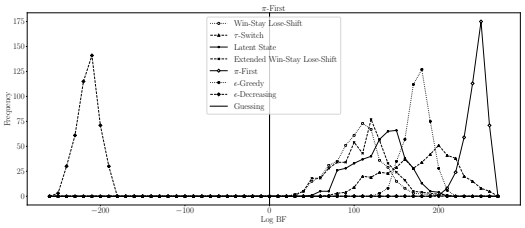
(c) e-WSLS



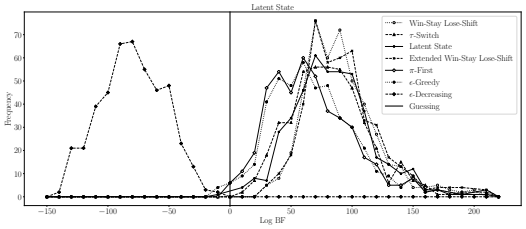
(d) ϵ-greedy



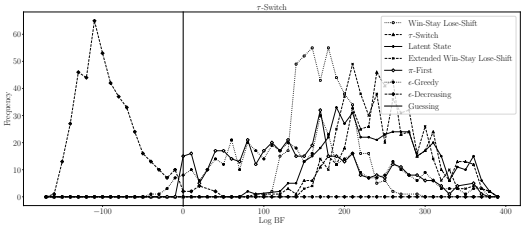
(e) ϵ-decreasing



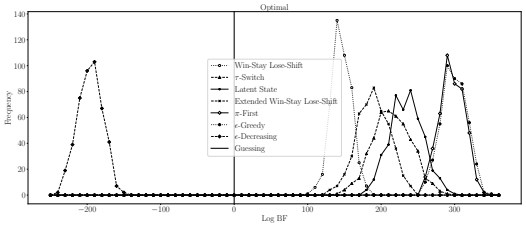
(f) π-first



(g) Latent State

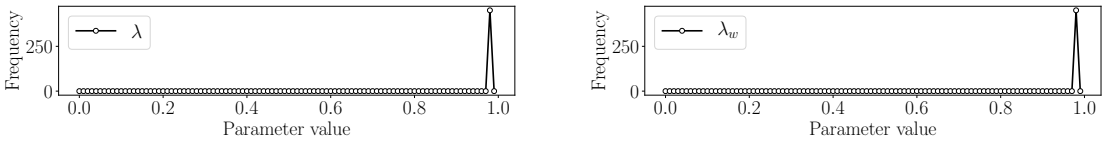


(h) τ-switch



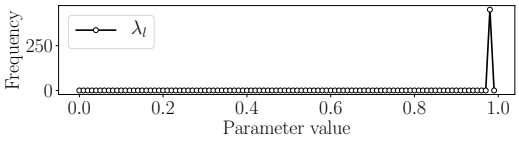
(i) Optimal

Figure A.2: Log BF measures for models applied to artificial datasets. Compare with (Laverty, 2019, Figure 5.1).

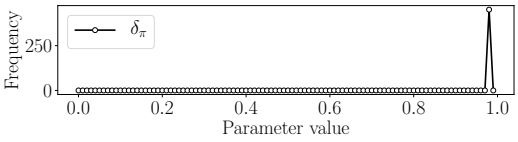


(a) λ (WSLS)

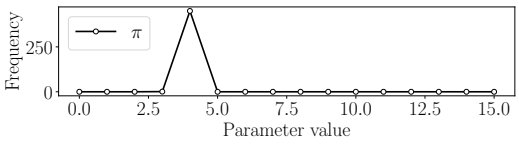
(b) λ_w (e-WSLS)



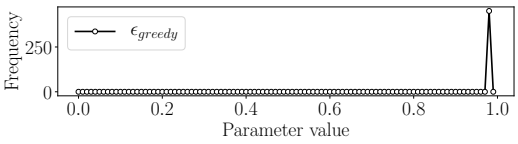
(c) λ_l (e-WSLS)



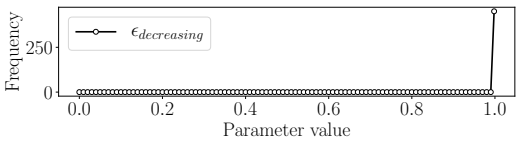
(d) δ (π-first)



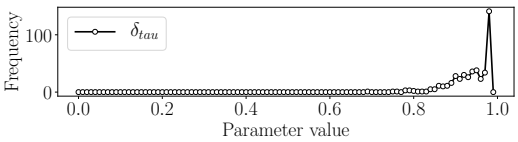
(e) π (π-first)



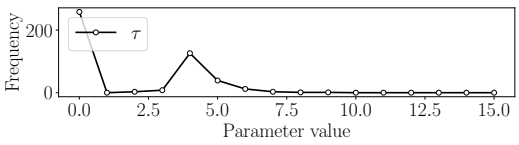
(f) ϵ (ϵ-greedy)



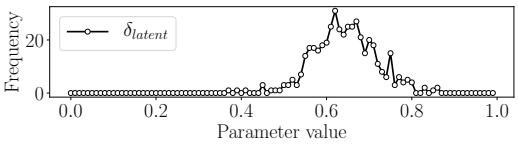
(g) ϵ (ϵ-decreasing)



(h) δ (τ-switch)



(i) τ (τ-switch)



(j) δ (Latent State)

Figure A.3: MAP Estimates on artificial data. Compare with (Laverty, 2019, Figure 5.2).

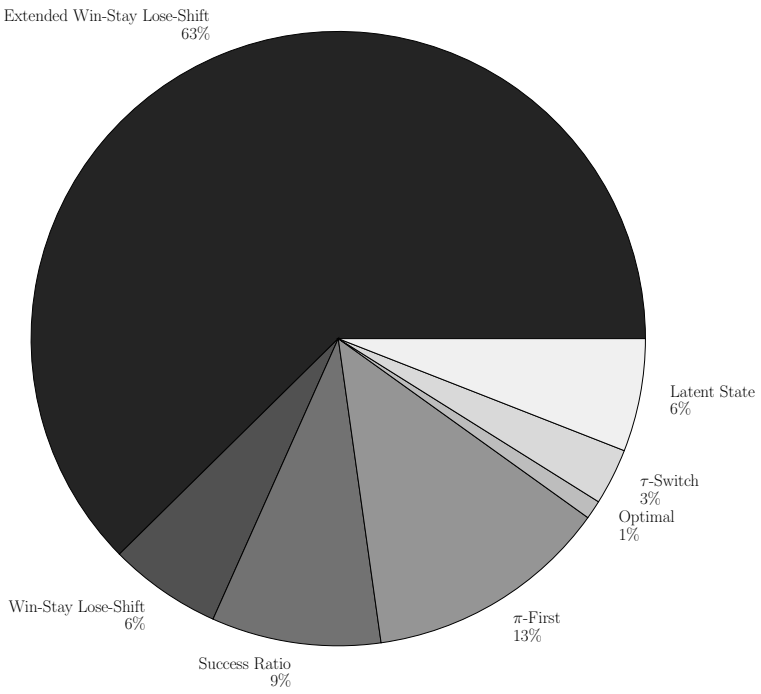
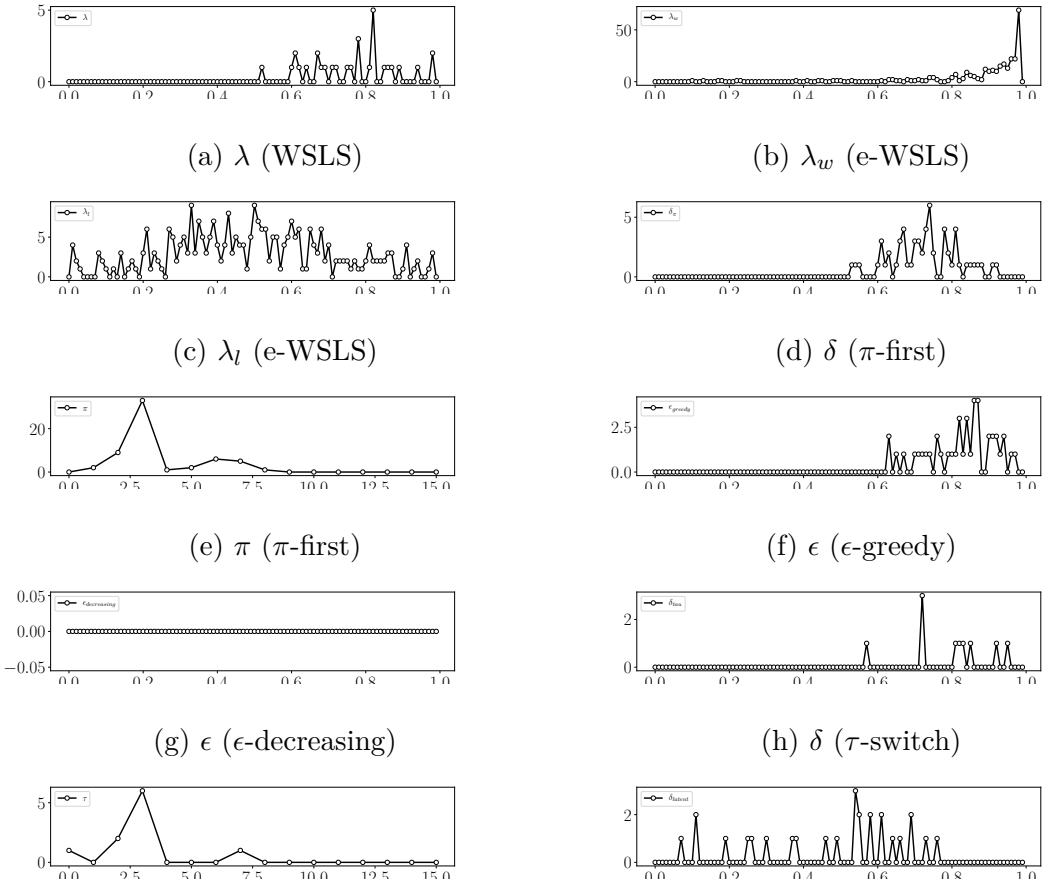


Figure A.4: Proportion of participants using each model according to maximum value of log BF measure. Compare with (Laverty, 2019, Figure 5.4).



(a) λ (WSLS)

(b) λ_w (e-WSLS)

(c) λ_l (e-WSLS)

(d) δ (π-first)

(e) π (π-first)

(f) ϵ (ϵ-greedy)

(g) ϵ (ϵ-decreasing)

(h) δ (τ-switch)

(i) τ (τ-switch)

(j) δ (Latent State)

Figure A.5: MAP Estimates on human data. Compare with (Laverty, 2019, Figure 5.5).