NTNU
Det skapende universitet

Statistical Models for the Beads Task Problem in Psychiatry

Benedicte Meløy Christensen

Master i fysikk og matematikk
Innlevert: juni 2016
Hovedveileder: Håkon Tjelmeland, MATH

Norges teknisk-naturvitenskapelige universitet
Institutt for matematiske fag

Statistical Models for the Beads Task Problem in Psychiatry

Benedicte Meløy Christensen

June 2016

MASTER THESIS

Industrial Mathematics

Department of Mathematical Sciences

Norwegian University of Science and Technology

Preface

This thesis is written as a part of my Master of Science degree at the Norwegian University of Science and Technology. It concludes the five-year study programme in "Applied Physics and Mathematics" with specialization in "Industrial Mathematics". I would like to thank my supervisor Håkon Tjelmeland for continually following up on my work and giving me guidance. I would also like to express my gratitude to Robert Biegler and Gerit Pfühl for introducing me to their work within psychology and for providing me with a data set.

Abstract

The tendency to "jump to conclusion" and make decisions on basis of little evidence has been linked with being prone to delusions. This behavior is particularly seen in the "beads task", where participant are presented with a sequence of beads and asked to decide which jar the beads are drawn from. In this thesis, we construct statistical models that explain the observed behavior in the beads task. Our models are based on work previously done by Moutoussis *et al.*. The models in our thesis contain quantifiable parameters that capture the bias towards making hasty decisions and the noise in decision-making. Two of the models also incorporate the noise in subjective probability estimates. We describe how the model parameters can be estimated through Bayesian analysis and perform a simulation study which shows that the parameters can be accurately retained. Finally, we fit the models to a real data set. These models may be utilized to see if there are any group differences between deluded patients and healthy controls.

Contents

1 Introduction **2**

1.1 Background 2

1.2 The Beads Task 6

2 Statistical Theory **9**

2.1 Hierarchical Bayesian Models 9

2.2 The Metropolis-Hastings Algorithm 11

2.3 MCMC Diagnostics 12

3 The Optimal Stopping Framework **15**

3.1 Ideal Bayesian Agent's Approach 15

3.2 Agent with Behavioral Uncertainty 20

3.3 Hierarchical Costed Bayesian Models 21

3.3.1 The HCB Model 23

3.3.2 The HCBU Model 24

3.3.3 Incorporating the Likelihood Ratings - The HCBP and HCBUP Models . . . 25

4 Parameter Estimation **29**

4.1 Maximum Likelihood Estimation 30

4.2 Bayesian Parameter Estimation 32

4.2.1 Choice of Hyperprior 32

4.2.2 Choice of Proposal Distribution for M-H 34

4.3 Parameter Estimation in the HCBP and HCBUP Models 37

5 Simulation Study **40**

5.1 M-H Simulations Under Different Scenarios 40

6 Data Set and Results **54**

6.1 The Raw Data 54

6.2 Model Results and Discussion 57

7 Conclusion and Future Work **63**

A Results **65**

A.1 HCBU Model 65

A.2 HCBP Model 65

A.3 HCBUP Model 65

B Acronyms **71**

Chapter 1

Introduction

The "beads in a jar task" is a psychology experiment that frequently is used in order to access how human beings make statistical inference. We describe this task in the coming section. It has been discovered that delusion-proneness is associated with making hasty decisions based on little evidence when performing this task. In this report, we construct four alternative models that describes the decision-making process in the beads task. The report is structured as follows: In the first chapter, we look at the broader context and explain why it is relevant to make statistical models that describe how human beings perform the beads task. The second chapter presents some statistical concepts that are needed later in the report. In the third chapter, we build four statistical models, while in the fourth chapter, we derive a procedure for estimating the parameters in these models. We test our parameter-estimation procedure on simulated data sets with known parameters in chapter five. In chapter six, we fit the models on actual data sets from human beings performing the beads task. We also discuss the most important findings. Finally, in chapter seven, we sum up our findings, discuss the limitations of our models, and present some suggestions for further work.

1.1 Background

Schizophrenia (SZ) is characterized by positive symptoms such as hallucinations and delusions, negative symptoms such as lack of motivation, and cognitive symptoms such as trouble focusing. Delusions are considered as one of the key traits of schizophrenia and it occurs in about three fourths of those diagnosed [9]. A delusion is a fixed belief held by a person in spite of emergence of strong conflicting evidence [1]. In order to improve treatments of mental disorders in which delusions are present, such as SZ, researchers have attempted to understand the psychological mechanisms behind delusions. This involves understanding how delusions are formed and how they are maintained [9]. A number of theories have been suggested [7]. It has been hypothesized that reasoning biases cause people to arrive at false conclusions which in turn causes them to form or maintain delusions. The word 'bias' in this context means the tendency to systematically behave in a way that differs from some reference group. Such reference groups may be healthy controls or psychiatric controls. In the following, we introduce several reasoning biases that researchers have suggested to be present in deluded patients. Some of these biases contradict each other, some elaborate on each other, and some compliment each other.

A reasoning bias that frequently has been reported is the tendency to making hasty decisions on the basis of little evidence. In literature, this reasoning bias is referred to as the *jumping to conclusion* (JTC) bias [7]. The JTC bias can explain formation of delusions because an implausible hypothesis may be prematurely accepted and therefore prevent more realistic alternatives from being considered. Probabilistic inference tasks, for example beads in a jar tasks, are often utilized to observe the JTC behavior. There are many versions of the beads task, but a typical version designed to capture the JTC bias goes as follows: Participants are told that beads will be drawn with replacement from one of two jars. Each jar contains black and white beads. One of the jars contains substantially more white than black beads and the other jar contains the opposite ratio of white to black beads. This ratio is known to the participants. The beads are drawn one by one from a hidden jar, and the participants can see another bead until they are certain about which of the jars the beads are being drawn from. All the participants are given the same predetermined sequences of beads, and the number of beads drawn before a decision is made is recorded. The participants are given several such tasks, and the ratios of black to white beads may vary from tasks to task. This version of the beads task is often called the *draws to decision* (DTD) beads task, as the main outcome is the number of beads drawn before a decision is made [7]. Numerous studies have concluded that delusion-prone individuals in fact collect less evidence than healthy controls before making a decision in the DTD beads task. A meta-analysis on the DTD beads task, based on 55 studies, has been performed by Dudley *et al.* [5]. They concluded that people with psychosis collect significantly less evidence then controls before making a decision in the two-jar DTD task.

Recent studies performed by Moritz *et al.* [12] challenge the idea that deluded patients systematically gather less evidence than controls before making a conclusion. They agree that a JTC bias is present in the two-jar DTD beads task, but they claim that deluded patients do not generally show a tendency to jump to a conclusion. Instead, they suggest the *liberal acceptance* (LA) bias. This account claims that deluded patients have a lower threshold for accepting an alternative as viable than controls; however, this does not necessarily imply a premature response since multiple alternatives may be considered plausible, and no single option can be selected. In the case of two jars, the LA bias explains the tendency to JTC because one of the options early surpass the threshold while the other option does not, causing the former to be chosen. Under ambiguity, however, the LA account predicts that the decision in fact may be delayed. For example, in a DTD beads task with four possible jars, the lowered threshold may cause two of the jars to be deemed plausible in situations where a higher threshold would only have deemed one as plausible. Here, the person with the high threshold is not ready to conclude, while the person with the low threshold needs more evidence [12]. Moritz *et al.* put the LA account to a test by letting deluded patients and healthy controls perform beads experiments with two jars and with four jars. Their conclusion was in agreement with the LA account: JTC bias was indeed present in the two-jar task while there was no group difference in the four-jar task. The LA account may explain why a delusion is formed: a low acceptance threshold may cause an unlikely hypothesis to be accepted as tenable.

Yet another reasoning bias that is postulated to be associated with delusion is the so called *overadjustment bias* or *bias toward disconfirmatory evidence* [15]. This bias is the tendency to be very responsive to disconfirmatory evidence, and it has been reported in the *draws to certainty* (DTC) version of the beads task. In the DTC version of the beads task, participants are shown a predetermined sequence of beads as in the DTD version. However, after each bead is drawn, participants are asked to provide a probability estimate for one of the jars being the source of the beads. A new bead is drawn until all the beads in the predetermined sequence are displayed. Some studies have found that deluded participants change the probability rating more than controls when for example a white bead is shown after a long streak of black beads. This is the bias toward disconfirmatory evidence; the participant reacts strongly when evidence against their currently favored hypothesis is presented. The overadjustment bias is in contradiction to another bias that frequently is reported, called the *bias against disconfirmatory evidence* (BADE). This is a bias where the individuals show a strong commitment to the initially favored hypothesis even when presented with evidence strongly supporting another hypothesis. This bias has been discovered in tasks where participants are presented with pictures from a comic strip, one by one. They are asked to interpret what is going on in the comic strip, and they are given several alternative interpretations to choose among. The first picture is designed to favor one or two of the available alternatives, while the subsequent pictures are strongly supporting one of the other alternatives. It has been discovered that deluded participants are less willing to move away from the interpretation that initially seemed plausible [16]. The LA account and the BADE may together explain why delusions are formed and maintained. While the LA account may explain why delusions are formed, BADE may explain why they are maintained; When an unlikely idea has been embraced, incoming evidence against this idea is for some reason resisted.

As noted earlier, delusions are beliefs that are held in spite of strong evidence contradicting them. However, the overadjustment bias, which has been reported in the beads task, says that deluded patients are easily persuaded by incoming disconfirmatory evidence. This bias seems to be in disagreement with the very nature of delusions, namely that beliefs are resistant to disconfirmatory evidence. Speechley *et al.* [15] have performed a study to gain more insight into the overadjustment bias, and they discovered what they refer to as the *hypersalience* account. This account may make the overadjustment bias compatible with delusions. Before we explain this account, we take a look at the experimental procedure that gave data supporting the account. Instead of the regular DTC procedure with one probability slider, Speechley *et al.* provided one slider for each of the jars. These sliders, or likelihood ratings, were on a continous scale ranging from "Very Unlikely" to "Very Likely". The experimental data showed that the deluded group had an exaggerated increase in likelihood ratings relative to the control group for whichever jar [1] that matched the current bead; however, there was no exaggerated decrease in the likelihood ratings for the jar that did not match the current bead. Based on this, Speechley *et al.* suggest that the overadjustment reported in other studies is not caused by an strong reaction to the disconfirmatory evidence – instead, they suggest that the overadjustment is a consequence of the deluded participants giving more faith in the option that is supported by the newest piece of evidence [7]. Putting greater trust in the hypothesis supported by the most recent piece of information is what Speechley *et al.* call the hypersalience account.

As we have seen, several cognitive biases that may explain delusions have been proposed. These biases have been supported by experimental data, often through variants of the beads task. The data analyses have mainly been done by calculating trivial summary statistics and then using for example ANOVA for testing whether there are group differences. Less attention has been directed towards making statistical models for the decision-making process in the beads task, containing key parameters that can be quantified. Moutoussis *et al.* [13], however, have presented one such statistical model. In this report, we introduce their model - as well as several tweaked versions of it - and estimate the models parameters based on a data set from a group of healthy subjects. The model aims to capture parameters that can explain the JTC behavior seen in the beads task. The goal of this report is to lead the way for building statistical models that describe the decision-making process in the beads task.

1.2 The Beads Task

As described in the previous section, the classical DTD beads task works as follows: There are two or more sources/jars of beads, each with a given ratio of white to black beads. The participants are told that beads will be drawn with replacement from one of the sources and that the source will remain the same throughout the trial. The color of new bead as well as the previously drawn beads are shown to the participant at all times. After each bead is drawn, the participant

[1] Instead of jars and beads, Speechley *et al.* used lakes and fish. Their computer program looks similar to the one shown in Figure 1.1.

Table 1.1: The trials and the proportion white (W), black (B) and red (R) beads in each of the jars. In the three first taks there are two possible jars. In the fourth task, there are four possible jars.

		Jar A		**Jar B**		**Jar C**		**Jar D**		
Task	**Sequence**	W	B	W	B	W	B	W	B	R
1	●●●○●●●●○●	0.2	0.8	0.8	0.2	-	-	-	-	-
2	○○○○●○○○○○	0.9	0.1	0.1	0.9	-	-	-	-	-
3	●○●○○●○●●○	0.5	0.5	0.8	0.2	-	-	-	-	-
4	○○○●○○○○○○	0.1	0.9	0.5	0.5	0.9	0.1	0	0.1	0.9

can either 1) state that he is sure about the source of the bead(s) and which jar this is or 2) state that he wants to see more beads before deciding. The participants are told that they can see a maximum of n beads. In the version of the beads task that we are studying in this paper, the participants additionally have to provide likelihood estimates for each of the jars being the source after each bead is drawn, as in the DTC task with one slider for each jar. This is done by having separate sliders ranging continuously from "Can't be this lake" to "Must be this lake" for each of the jars. Prior to the experiment, the participants are instructed that even though they have decided on a jar, new beads will be drawn and probability estimates must be provided until all the $n = 10$ beads are drawn. They do not have to commit to any jar during the sequence; when 10 beads have been drawn, they can state that they still are not sure about which of the jars that is the source. The experiment is performed on a computer. Instead of jars and beads, the participants are confronted with lakes and fish. A screen shot of the interface is displayed in Figure 1.1.

Each participant is given four tasks. These are summarized in Figure 1.1. The tasks are exactly the same for all the participants. In the first three tasks, there are only two possible sources of beads. In the last task however, there are four possible sources. More specifically, in Task 1, jar A has 20% white beads and 80% black beads while jar B has 80% white beads and 20% black beads. The sequence "BBBWBBBBWB" is given to the participants, where "W" and "B" indicate white and black beads, respectively. This task is the same as task number one in [12]. In Task 2, jar A has 90% white and 10% black beads while jar B has 10% white and 90% black beads. The sequence provided is "WWWWBWWWWW". This is the same as task number two in [12]. In Task 3, jar A has 50% white and 50% black beads while jar B has 80% white and 20% black

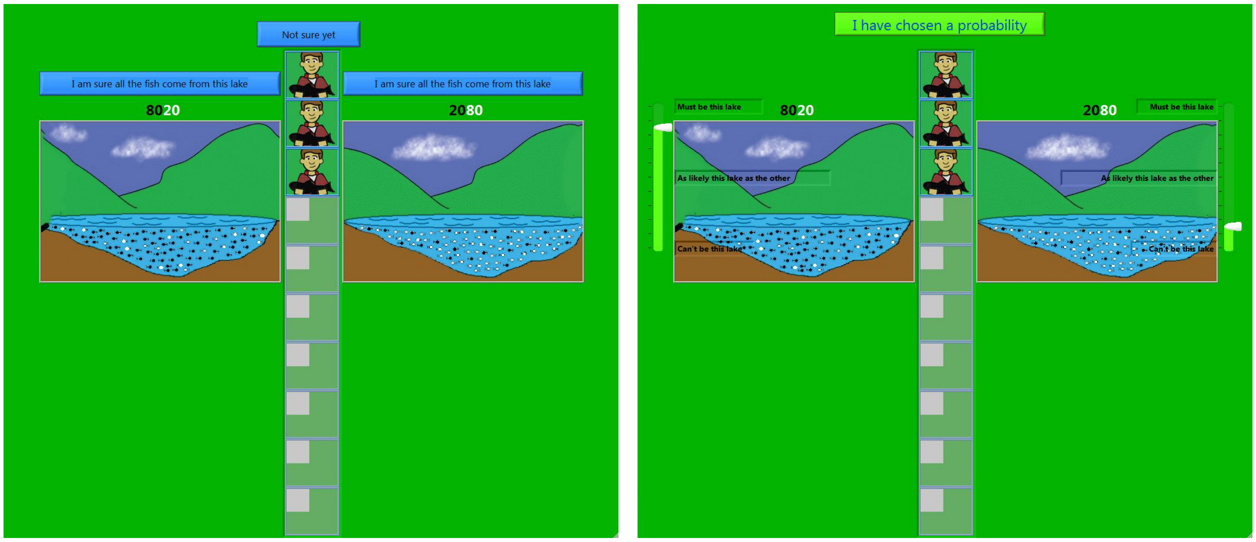
Not sure yet
I am sure all the fish come from this lake
I am sure all the fish come from this lake
8020
2080
I have chosen a probability
Must be this lake
8020
2080
Must be this lake
As likely this lake as the other
As likely this lake as the other
Can't be this lake
Can't be this lake

Figure 1.1: Two screen shots from the experiment. In the left panel, the participant can choose to commit to one of the jars or to postpone the commitment. No matter which of these actions the participants chooses, he is asked to provide likelihood estimates for each of the jars. This is shown in the right panel. Source: Robert Biegler.

beads, and the sequence "BWBWWBWBBW" is drawn. This is the same task as task number one in [15]. In the last task, there are four possible sources of beads instead of only two. Jar A contains 10% white and 90% black beads, jar B contains 50% white and 50% black beads, jar C contains 90% white and 10% black beads and jar D contains 10% black and 90% red beads. The sequence "WWWBWWWWWW" is provided. This is the same as task number three in [12]. Researchers at the Department of Psychology at respectively the "Norwegian University of Science and Technology" and the "Arctic University of Norway" have designed this experiment and gathered experimental data.

Chapter 2

Statistical Theory

In this chapter, we introduce some statistical concepts that are utilized in this report. In particular, we discuss hierarchical Bayesian mode, the Metropolis-Hastings algorithm and Markov chain Monte Carlo diagnostics.

2.1 Hierarchical Bayesian Models

Hierarchical models or *multilevel models* are useful in statistical applications where the is believed to be some dependency between the parameters. Figure 2.1 shows a typical hierarchical structure. Let us use this figure and an example to illustrate the idea behind hierarchical models. For instance, let us say we have n individuals, and each of these individuals are characterized by some parameter τ_i, where i denotes the index of the individual. This parameter is not observed itself, but there is a known stochastic model that maps τ_i to some observed output of the individual, y_i. That is, the density function $p(y_i|\tau_i)$ is known or specified. The observable output constitutes the first level of the hierarchical structure. The unobserved individual-parameters τ_i make up the second level of the model. Let us say that all the individuals are expected to be somewhat similar. We can model this by imposing a common prior distribution on the parameters τ_i, namely a *population distribution* [10]. The population distribution is specified by some *hyper-parameter*, let us call it α. In other words, we have the density function $p(\tau_i|\alpha)$ which is common for all i's. At this point, the model is specified such that the likelihood $p(y_1,...,y_n|\alpha)$ can be expressed by integrating over the τ_i parameter for each individual. In order for this model to

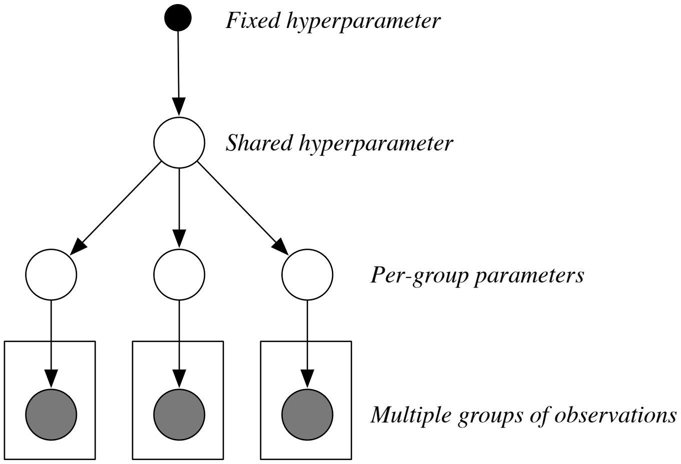

Figure 2.1: A typical hierarchical structure. Credit: David M. Blei.

be fully Bayesian, however, we need to view the hyperparameter as a random variable by putting a prior on it, i.e. a *hyperprior* $p(\alpha)$. The parameters of the hyperprior are known and they make up the top level of the hierarchical structure. Then, we have a probability model on the entire set of parameters. The joint posterior distribution of all the model parameters can be written out as

$$p(\tau_1,...,\tau_n,\alpha|y_1,...,y_n) = \frac{p(\alpha)\prod_{i=1}^{n} p(\tau_i|\alpha)p(y_i|\tau_i)}{p(y_1,...,y_n)} \tag{2.1}$$

After having obtained the posterior distribution in a hierarchical model, it may be desirable to look at some elements that summarize its information. One may for example be interested in location parameters, e.g. median and mode, and dispersion parameters, e.g. variance and precision [8]. These measures give us information about the parameters in the model. We can look at each individual in the group by investigating the measures concerned with the marginal distribution of the τ_i's, but we can also say something about the group as a unit by looking at the meaures concerned with the marginal distribution of the hyperparameter α.

If one believes that there is some similarity across the individuals, constructing a Bayesian hierarchical model is advantageous. A hierarchical structure provides *borrowing strength* across individuals. This means that observation from one individual contributes with information to the individual-specific parameter of another individual, through the hyper-parameter. The individual-specific parameters are shrunken towards a common mean, and this common mean is itself driven by the observations. As a consequence, we may make better inference about the individual-parameters than in the case where the individual-specific parameters are based only on the observations from the individual in question. This is particularly advantageous if there are few observations per individual.

2.2 The Metropolis-Hastings Algorithm

Markov chain Monte Carlo (MCMC) methods are powerful tools for sampling from high-dimensional probability distributions. These methods rely on constructing a Markov chain whose stationary distribution is the *target* density function that we wish to sample from. The Monte Carlo part refers to utilizing random sampling to obtain empirical result, while the Markov chain part refers to the fact that the sample is obtained by simulating a Markov chain. MCMC methods are useful for numerically calculating high dimensional integrals that are difficult to solve analytically, for example the expected value or variance of a multidimensional random variables [8].

The Metropolis-Hastings (M-H) algorithm is a MCMC procedure that we utilize in this report. We do not go into the theory behind the algorithm, but instead we introduce how the algorithm works and refer to [8] for a more profound introduction. Let us denote the target density function from which we want to sample as $\pi(\boldsymbol{\theta})$. This may for example be the posterior density function in (2.1). Also, let $\boldsymbol{\theta}^{(t)}$ be the t-th value sampled by the M-H algorithm. The algorithm goes as follows: First, choose an arbitrary[1] initial state for the chain, $\boldsymbol{\theta}^{(0)}$. Also, choose an arbitrary *proposal distribution* $q(\cdot|\boldsymbol{\theta}^{(t)})$ from which the candidate states will be sampled, given the current state. The only requirement for the proposal distribution is that is must ensure that the resulting Markov chain is aperiodic and irreducible, as these are sufficient conditions for convergence to a unique stationary distribution [11]. Then, for each iteration t, sample a candidate value $\tilde{\boldsymbol{\theta}}$ from the proposal distribution $q(\cdot|\boldsymbol{\theta}^{(t)})$. Calculate the *Metropolis-Hastings Ratio,* $R(\tilde{\boldsymbol{\theta}}|\boldsymbol{\theta}^{(t)}) = \frac{\pi(\tilde{\boldsymbol{\theta}})q(\boldsymbol{\theta}^{(t)}|\tilde{\boldsymbol{\theta}})}{\pi(\boldsymbol{\theta}^{(t)})q(\tilde{\boldsymbol{\theta}}|\boldsymbol{\theta}^{(t)})}$. Set $\boldsymbol{\theta}^{(t+1)} = \tilde{\boldsymbol{\theta}}$ with probability $\alpha = \min\{1, R(\tilde{\boldsymbol{\theta}}|\boldsymbol{\theta}^{(t)})\}$, called the accep-

[1] The initial state must be chosen such that $\pi(\boldsymbol{\theta}^{(0)}) > 0$

tance probability. If the suggested parameter is not accepted, set $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)}$. The draws $\boldsymbol{\theta}^{(t)}$ become increasingly close to being draws from the limiting distribution $\pi(\boldsymbol{\theta})$ as t gets larger [8].

In the case where $\boldsymbol{\theta}$ is a vector, one can use component-wise updating, which means that only one of the components of the parameter vector $\boldsymbol{\theta}$ may change value at each iteration while the other components remain as they were in the previous time step. One can either cycle through the components of the parameters vector or choose a component randomly weighted by how often we would like the different components to be sampled [8]. This is a special case of block updating.

2.3 MCMC Diagnostics

In MCMC procedures, the draws become increasingly close to draws from the stationary distribution as t increases [8]. In order to obtain a sample that represents the target distribution, the sample should be taken from the chain after it has *converged* to its equilibrium distribution [8]. The word converged in this context means that the draws produced by the chain approximates the target density well enough. A poorly chosen starting point for the chain may cause the chain to take long to converge. For that reason, it is crucial to know whether the chain has converged. There exists some theory on how to obtain quantitative bounds on number of iterations needed for convergence, but it has had little impact on practical work [4]. Instead, one usually has to rely on diagnostic tools to get some idea of whether convergence is reached. This involves applying graphical techniques and calculating statistical properties of the output of the chain.

The samples generated by a MCMC algorithm are dependent by construction, since the next state in the chain depends on the current state. The *mixing* of a chain is concerned with how far apart in the chain two sampled values must be in order to be considered approximately independent [11]. If the values that are sampled by the MCMC procedure are strongly correlated, the sample needs to be large in order to represent the target density well. The mixing of a chain has a say in how quickly the chain forgets its starting value and explores the support of the target density [11]. For this reason, the rate of convergence in a MCMC algorithm is affected by the mixing property of the chain. It is common practice to discard some *burn-in period* in the beginning of the chain in order to exclude the draws that are generated before convergence is reached.

As noted above, obtaining a good sample through MCMC simulations requires the user to have control of two properties of the chain, namely the rate of convergence and the mixing. The first of these governs the burn-in period, and the latter governs the the run-length (and the burn-in period). The user must 1) have an idea of whether convergence is reached so that representative values can be collected, and 2) know how good the mixing of the chain is, so that he knows how large the sample should be. The rate of convergence and the mixing of a chain are properties that overlap. Therefore, many techniques can be used to access both these properties at once. In the following, we introduce some of these techniques.

The *trace plot* shows the sample number t plotted against a component of the sampled value $\boldsymbol{\theta}^{(t)}$. These plots can provide a hint of whether convergence is reached, as convergence is characterized by rapid fluctuation around a stable mean [3]. Bad mixing can also be revealed through a trace plot. A chain that is mixing poorly maintains the same - or nearly the same - values over many successive iterations [11]. Another graphical tools that may be useful in reveling potential problems with MCMC algorithm is *auto-correlation plots*. This plots shows the correlation in the sequence of $\boldsymbol{\theta}^{(t)}$ at different lags. If $\boldsymbol{\theta}^{(t)}$ is a vector, the autocorrelation-plot can be plotted for each parameter. A slow decay may suggest bad mixing [11]. Monitoring the *cross-correlation* of the chain, namely the correlation between parameters, may also be useful . If two parameters are strongly correlated, it may suggest a poor choice of parameterization or overparameterization [4] and that the convergence is slow. The cross-correlations can be presented through scatter plots of pairs of parameters in the parameter vector $\boldsymbol{\theta}^{(t)}$. It can also be visualized through a correlogram, which is a image of the correlation matrix.

For the Metropolis-Hastings algorithm, the *acceptance rate* of the suggestions from the proposal distribution should be monitored. This is because the acceptance rate says something about whether the spread of the proposal distribution is appropriate, and the spread of the proposal influences the mixing and the convergence of the chain. A spread that is too large causes many proposals to be rejected and the chain will move slowly because the same state is visited many times in a row (causing slow convergence), but a spread that is too small will cause the chain to explore the support of the target distribution slowly because each step taken in the chain is small [2]. Literature suggest an acceptance rate between 20% and 50% as a rule of thumb [8].

There exists a number of convergence diagnostics that are more formal and that do not rely on visual inspection, e.g. the Gelman and Rubin diagnostics. However, for the analysis in this report, the visual techniques introduced above will suffice. We refer to [4] for an introduction of other convergence diagnostics.

Chapter 3

The Optimal Stopping Framework

The DTD experiment is a task in which the agent has to determine when to stop collecting evidence and instead make a decision about the source of the beads. This can be viewed as an *optimal stopping problem*. When we talk about the *optimal strategy* for this problem, we mean the strategy that *maximizes the expected utility*. So what is the utility in the DTD task? In the experiment instructions, no information is provided about the reward (r) of answering correctly, the cost of answering incorrectly (c^w), nor the cost of sampling another bead (c^s). However, we may assume that each participant has his own internal values for these parameters. For example, some participants may be very concerned about answering incorrectly, and thus have a large reward for correct answers, large cost for incorrect answers and small cost of sampling. The utility in this framework is the units gained by answering correctly minus the units spent on sampling and answering incorrectly. Given a set of parameters (r, c_w, c^s) there is an optimal strategy for the DTD task. The optimal stopping framework for the DTD problem is introduced by Moutoussis *et al.* in [13]. In the following sections we present, elaborate on and make some modifications to his work.

3.1 Ideal Bayesian Agent's Approach

In this section, we introduce the optimal strategy for the DTD problem. We call an agent following this strategy an ideal Bayesian agent (IBA). We provide detailed derivations for the case where there are only two possible sources of beads, and where these sources contain white and black beads only. However, the calculations are similar for the case where there are more than two sources and more than two colors. We define the rules to be such that the agent has to commit to a jar at some point during the course of the ten beads. This is different from the rules given to the participants in the data set that we study; however, in Section 6.2 we describe how we can handle this difference. Let r_A and r_B be the proportions of white beads in jar A and B, respectively. Furthermore, let x_i denote the color of the i^{th} bead drawn, taking the value 0 if the bead is black and 1 if the bead is white. Furthermore, let $\mathbf{x} = \{x_1, ..., x_n\}$ be the whole sequence given in the task. Then, the set $\{\mathbf{x}, r_A, r_B\}$ defines the task. Given the parameters (r, c^w, c^s), the task has a optimal solution, namely the pair (d, m), where $d \in \{d^A, d^B\}$ is the urn chosen and $m \in \{1, ..., n\}$ is the number of beads displayed before deciding on the jar. Let us denote the solution strategy of an individual as $\mathbb{S}$, and in particualar, let $\mathbb{S}_{IBA}$ be the solution strategy of the IBA. Furthermore, let us define the utility (U) as the reward (R) minus the cost (C) in the task, namely

$$U = R - C. \tag{3.1}$$

Next, let us define a *state* as $s_i = n_w$ for $0 \leq n_w \leq i \leq n$, where i is the total number of beads drawn and n_w is the number among these that are white. Since only white or black beads are drawn, it is implicitly given that the number of black beads are $i - n_w$. Also, let d_i denote the *action* taken by the agent after the i^{th} bead is drawn. d_i can take on values among d^A (decide jar A), d^B (decide jar B), and d^S (decide to sample one more bead) depending on how many beads i that have been drawn; d^S is only an available choice when strictly less than n beads have been drawn. Let $\mathbb{D}_i$ denote the set of available actions for a given task when i beads have been drawn. Now, we define the *action value*, Q, for taking an action d in the state $s_i = n_w$ under strategy $\mathbb{S}$ as the expected additional utility of taking action d in state $s_i = n_w$ under strategy $\mathbb{S}$; that is,

$$Q(d, i, n_w, \mathbb{S}) = E[U | S_i = n_w, D_i = d, \mathbb{S}]. \tag{3.2}$$

The optimal strategy for the DTD problem, $\mathbb{S}_{IBA}$, is to calculate the action values in state $s_i = n_w$ (knowing that you behave as an IBA) and deterministically choose the action that has the largest action value. This is the strategy that in the long run will yield the largest gain (or smallest cost). We proceed by deriving the expression for the action values in (3.2) under the IBA strategy. We start by looking at the action values for deciding on each of the jars. Let $V \in \{A, B\}$ be the jar that is being drawn from. We use the letter V for vase. Then, the action value for choosing jar $v \in \mathbb{V} = \{A, B\}$ in any state $s_i = n_w$ can be written as

$$\begin{aligned} Q(d^v, i, n_w, \mathbb{S}) &= E[U|S_i = n_w, D_i = d^v, \mathbb{S}] \\ &= E[U|S_i = n_w, D_i = d^v] \\ &= r \cdot P(V = v|S_i = n_w) - c_w \cdot P(V \neq v|S_i = n_w) \\ &= r \cdot P(V = v|S_i = n_w) - c_w \cdot (1 - P(V = v|S_i = n_w)) \\ &= (r + c_w)P(V = v|S_i = n_w) - c_w. \end{aligned} \tag{3.3}$$

We note that the action value for choosing a jar does not depend on the strategy of the agent. Let us take a look at the posterior probability of the source being jar v, which is the second term in the expression above. In the experiment, the prior probability for each of the jars are equal, so we have $P(V = A) = P(V = B) = \frac{1}{2}$ in task number one through three. Let r_v be the ratio of white beads in jar v. Then, we have that the probability of the source of beads being jar $v \in \mathbb{V} = \{A, B\}$, when $i \in \{1, .., n\}$ beads are drawn and $n_w \in \{0, ..., i\}$ of these are white, is

$$\begin{aligned} P(V = v|S_i = n_w) &= \frac{P(V = v) \cdot P(S_i = n_w|V = v)}{P(S_i = n_w)} \\ &= \frac{P(V = v) \cdot P(S_i = n_w|V = v)}{\sum_{\tilde{v} \in \mathbb{V}} P(V = \tilde{v}) \cdot P(S_i = n_w|V = \tilde{v})} \\ &= \frac{P(S_i = n_w|V = v)}{\sum_{\tilde{v} \in \mathbb{V}} P(S_i = n_w|V = \tilde{v})} \\ &= \frac{r_v^{n_w}(1 - r_v)^{i - n_w}}{\sum_{\tilde{v} \in \mathbb{V}} r_{\tilde{v}}^{n_w}(1 - r_{\tilde{v}})^{i - n_w}}, \end{aligned} \tag{3.4}$$

where we in the third equality used that the prior probabilities for each jar are equal. Equations (3.3) and (3.4) are what we need in order to calculate the action values for choosing jar A and jar B in any given state. Now, let us move on to deriving the expression for the action value of sampling another bead. This action value depends on the strategy of the agent, since the expected future return depends on the behavior of the agent. As we will see, the agent must "search" through all the future outcomes to calculate this quantity. Define the probability of choosing action d in state $s_i = n_w$ under the strategy $\mathbb{S}$ as the weight

$$w(d,i,n_w,\mathbb{S}) = P(D_i = d|S_i = n_w,\mathbb{S}). \tag{3.5}$$

The IBA agent deterministically chooses the action with the largest action value, so all the weight is given to the action with the largest action value, and we have

$$w(d,i,n_w,\mathbb{S}_{IBA}) = \begin{cases} 1 & \text{if } d = \underset{a\in\mathbb{D}_i}{\arg\max}\, Q(a,i,n_w,\mathbb{S}_{IBA}), \\ 0 & \text{otherwise.} \end{cases} \tag{3.6}$$

If there are two or more actions with equal action values and these are larger than all the other action values, it does not matter which of these the an agent chooses in order to be ideal; therefore, we define the IBA to choose randomly among the two jars if the action values for these are equal and larger than the action value for sampling. If the action value for choosing one of the jars is equal to the action value for sampling, and this value is larger than the action value for choosing the other jar, we define the IBA to decide on the jar instead of sampling. Thus, the IBA will choose as quickly as possible. Next, we note that the expected utility of an agent in state $s_i = n_w$ given that the true source is jar $v \in \{A,B\}$ can be written as

$$\begin{aligned} E[U|S_i = n_w, V = v, \mathbb{S}] &= \sum_{d\in\mathbb{D}_i} P(D_i = d|S_i = n_w, V = v, \mathbb{S}) \cdot E[U|S_i = n_w, V = v, D_i = d, \mathbb{S}] \\ &= \sum_{d\in\mathbb{D}_i} P(D_i = d|S_i = n_w, \mathbb{S}) \cdot E[U|S_i = n_w, V = v, D_i = d, \mathbb{S}] \\ &= \sum_{d\in\mathbb{D}_i} w(d,i,n_w,\mathbb{S}) \cdot h(i,n_w,v,d,\mathbb{S}), \end{aligned} \tag{3.7}$$

where we have defined the function $h(\cdot) = E[U|S_i = n_w, V = v, D_i = d, \mathbb{S}]$ because it later will turn up recursively in the expression of the action value of sampling. We have

$$\begin{aligned} h(i,n_w,v,d,\mathbb{S}) &= E[U|S_i = n_w, V = v, D_i = d, \mathbb{S}] \\ &= \begin{cases} r \cdot I_{[v=A]} - c_w \cdot I_{[v\neq A]} & \text{if } d = d^A, \\ r \cdot I_{[v=B]} - c_w \cdot I_{[v\neq B]} & \text{if } d = d^B, \end{cases} \end{aligned}$$

and for the case where $d = d^S$, we have

$$
\begin{aligned}
&h(i, n_w, v, d^S, \mathbb{S}) \\
&= E[U|S_i = n_w, V = v, D_i = d^S, \mathbb{S}] \\
&= -c^s + \sum_{s\in\{n_w, n_w+1\}} P(S_{i+1} = s|V = v, S_i = n_w, D_i = d^S, \mathbb{S}) \cdot E[U|V = v, S_i = n_w, D_i = d^S, S_{i+1} = s, \mathbb{S}] \\
&= -c^s + \sum_{s\in\{n_w, n_w+1\}} P(S_{i+1} = s|S_i = n_w, V = v) \cdot E[U|V = v, S_{i+1} = s, \mathbb{S}] \\
&= -c^s + \sum_{s\in\{n_w, n_w+1\}} P(S_{i+1} = s|S_i = n_w, V = v) \sum_{d\in\mathbb{D}_{i+1}} P(D_{i+1} = d|V = v, S_{i+1} = s, \mathbb{S}) E[U|V = v, S_{i+1} = s, D_{i+1} = d, \mathbb{S} \\
&= -c^s + \sum_{s\in\{n_w, n_w+1\}} P(S_{i+1} = s|S_i = n_w, V = v) \sum_{d\in\mathbb{D}_{i+1}} w(d, i+1, s, \mathbb{S}) \cdot h(i+1, s, v, d, \mathbb{S}). \qquad (3.8)
\end{aligned}
$$

Here, we see a recursive pattern; in every state where there is an option of sampling another bead, the function $h(i, n_w, v, d^S, \mathbb{S})$ makes use of $h(i+1, n_w, v, d^S, \mathbb{S})$ and $h(i+1, n_w+1, v, d^S, \mathbb{S})$. This recursion terminates when n beads are drawn, because at this point it is not possible to sample another bead. The agent is forced to choose a jar. When $i = n$ in (3.7), d^S is no longer contained in the sum, so the recursive part stops.

Finally, the action value for sampling another bead in state $s_i = n_w$ where $0 < n_w \le i < n$ can be written out as

$$
\begin{aligned}
Q(d^S, i, n_w, \mathbb{S}) &= E[U|S_i = n_w, D_i = d^S, \mathbb{S}] \\
&= \sum_{v\in\mathbb{V}} P(V = v|S_i = n_w, D_i = d^S, \mathbb{S}) \cdot E[U|S_i = n_w, D_i = d^S, V = v, \mathbb{S}] \\
&= \sum_{v\in\mathbb{V}} P(V = v|S_i = n_w) \cdot h(i, n_w, v, d^S, \mathbb{S}). \qquad (3.9)
\end{aligned}
$$

The optimal stopping strategy for an agent that assumes the game parameters (r, c_w, c^s) is to calculate the action values $Q(d, i, n_w, \mathbb{S}_{IBA})$ for each of the available actions d, and then to choose the action that yields the largest value. This model is in fact a one-parameter model. We can fix two of the parameters, and let the last one be flexible. We choose $r = 0$ and $c^w = 100$ as a reference for the model.

3.2 Agent with Behavioral Uncertainty

Human beings do certainly not follow the strategy of the IBA. First of all, the strategy requires heavy computations that involve searching through all possible future states. It is unlikely that human beings are able to perform these. Secondly, humans do not choose as deterministically as an ideal agent; provided with the same information several times, the choices may still vary. To account for the behavioral uncertainty, we need to introduce some noise in the model. This can be done by for example assuming that the human agents pick an action randomly, but that the probability of picking each action is weighted based on the magnitude of the action value. We use the same weight function that is used in [13], namely the the Softmax function with an individual-specific parameter τ that specifies how arbitrary the choices tend to be. With Softmax weighting, the probability of choosing action d in state $s_i = n_w$ becomes

$$w(d, i, n_w, \mathbb{S}) = \frac{e^{Q(d,i,n_w,\mathbb{S})/\tau}}{\sum_{\tilde{d}\in\mathbb{D}_i} e^{Q(\tilde{d},i,n_w,\mathbb{S})/\tau}}. \tag{3.10}$$

We can see that as τ goes to zero, the probability of choosing the action with the largest action value goes to 1, and the agent gets closer to being an IBA. On the other hand, as τ goes to infinity, the probabilities for each of the actions become equal, namely one divided by the number of possible actions, and the agent makes arbitrary choices.

As we can see in (3.10), the probability of some agent choosing an action d depends on the action values for choosing the different actions. The action values for choosing actions d^A and d^B are calculated straightforwardly as shown in (3.3). Calculating the action value for sampling another bead, however, requires the agent to iterate through all possible future outcomes. These outcomes are not only related to what color the future beads have - they are also related to what action the agent chooses to perform when the beads appear. Therefore, in order to calculate the action value for sampling, the agent must have some idea - which may be correct or incorrect - of how he behaves. That is, he must have an idea of his own strategy, which may or may not coincide with his true strategy. Under behavioral uncertainty, the agent can either 1) know that he makes choices according to the softmax choice function or 2) think that he behaves as an IBA. Therefore we get two alternative models. In the first model, the agent takes into account that he in fact behaves stochastically, when calculating the action value for sampling another bead. Then, the choice is made stochastically based on the action values. We can refer to this model as the CB model, for Costed Bayesian model. In the second model, the agent calculates the action values for sampling as if he were an IBA, and then an action is chosen stochastically based on the weight of each action. We can all this the CBU model, for Costed Bayesian Unaware model, since the agent is unaware of his own decision noise. We stress that these models do not use the probability estimates provided by the participants in any manner. The only observations taken into account are the DTD's and the final jar chosen. In Section 3.3.3, we incorporate the subjective probability estimates into the framework already introduced above. When incorporating these estimates, we will arrive at two new models for how an agent acts in the beads task. We call these the CBP and the CBUP models. For now, however, we omit the subjective probability estimates in the models.

In the model(s) described above, each agent has the two parameters c^s and τ. The sampling cost parameter captures the eagerness to stop collecting evidence and instead make a decision. It may be seen as an equivalent to a decision threshold; a large sampling cost corresponds to a small decision threshold and vice versa. The noise parameter accounts for the randomness in the decision-making. If each individual provides responses for a large number of sequences, we are able to accurately estimate the specific parameters for each individual, for example through maximum likelihood estimation. However, for the data that we analyze in this report, there are only a few trials for each participant. As a consequence, there will be a large uncertainty associated with each individual's parameter estimates. We can alleviate this issue by constructing a hierarchical model, as done by Moutoussis *et al.* In the next section, we describe this model.

3.3 Hierarchical Costed Bayesian Models

In line with Moutoussis *et al.*, let us call the sampling costs (c_k^s) and noise parameters (τ_k) *micro-parameters*. These are individual-specific parameters. Within one population[1], it is reasonable to assume that the individuals are somewhat similar, so we expect some similarities between

[1] A population may for example be a group of schizophrenic individuals or a group of healthy individuals.

the individuals' micro-parameters. Thus, we can view the pairs of of micro-parameters (c_k^s, τ_k) as stochastic variables drawn independently from some common prior distribution governed by some unknown *hyper-parameters*. This hierarchical structure allows us to investigate the hyper-parameters instead of - or in addition to - the micro-parameters.

In the hierarchical structure, the sampling costs and noise parameters are considered to be drawn from some prior. As done by Moutoussis *et al.*, we can assume that the sampling costs are drawn independently from the gamma distribution with parameters α and β and that the noise parameters are drawn independently from the gamma distribution with parameters η and δ, namely

$$\begin{aligned} &c_k^S \sim \text{Gamma}(\alpha, \beta), \\ &\tau_k \sim \text{Gammma}(\eta, \delta), \quad k = 1, ..., N_{\text{ind}}, \end{aligned}$$

where N_{ind} is the number of individuals. Thus, the expected value of the sampling cost is $\mu_1 = E[c_k^s] = \alpha\beta$ and the standard deviation (SD) is $\sigma_1 = SD(c_k^s) = \sqrt{\alpha}\beta$. Similarly, for the noise parameters, we have $\mu_2 = \eta\delta$ and $\sigma_2 = \sqrt{\eta}\delta$. Under this hierarchical structure, we can estimate the hyperparameters α, β, η, and δ to infer about the population as a whole. The hierarchical structure is illustrated in Figure 3.1.

As noted in the previous section, the optimal stopping problem with behavioral uncertainty gives rise to two alternative models. We get the first model by assuming that the agents themselves know that they choose between the actions based on Softmax weights instead of choosing deterministically. In this case, the agent will calculate the action values for sampling another bead, knowing that he behaves stochastically. The agent is behaving ideally in the sense that he maximizes the expected reward in the setting where there is decision noise present. This is the same model as the one introduced in [13]. We will from now on refer to this model as the HCB model, for Hierarchical Costed Bayesian model. We get the second model by assuming that the agents are unaware of the fact that they behave stochastically, when calculating actions values for sampling. In this case, the agents calculate the action values as if they were IBA's but

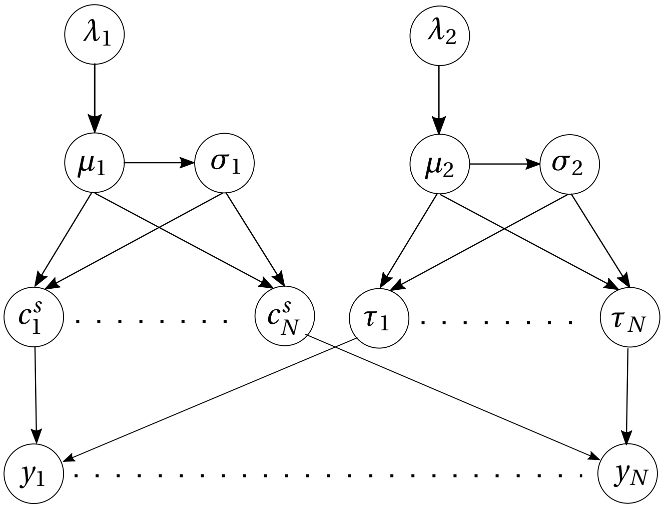

Figure 3.1: The structure of the HCB and the HCBU models. Instead of considering the parameters α, β, η, and δ, we consider μ_1, σ_1, μ_2, and σ_2. The constants λ_1 and λ_2 are concerned with the hyper-prior that is used in the Bayesian analysis discussed in section 4.2.

still choose randomly among the actions based on Softmax weights. This model will be referred to as the HCBU model, which is short for Hierarchical Costed Bayesian Unaware. Let us take a closer look at these two models.

3.3.1 The HCB Model

In the HCB model, the agent is aware of the fact that he has decision noise when choosing an action. Therefore, when he calculates the action value of sampling (i.e. the expected additional gain if he chooses to sample), he takes into account his stochastic behavior in the future. This means that he uses (3.10) when calculating (3.8). Consequently, his action value for sampling will be lower than an IBA's action value for sampling. As τ increases, the action value for sampling will decrease while the action values for each of the jars remain the same. Therefore, the probability of choosing to sample will decrease as τ increases, and it is converging to one over the number of possible actions. Thus, we see that increasing noise implies a greater probability of committing to a jar. The probability of committing to the jar that is posteriorly more likely of being the source is always greater than the probability of committing to any of the other jars. We know that the action value of sampling increases as c^s increases. As a result, early decisions in this model can be caused by having a large c^s and/or by having a large τ. If there are a great

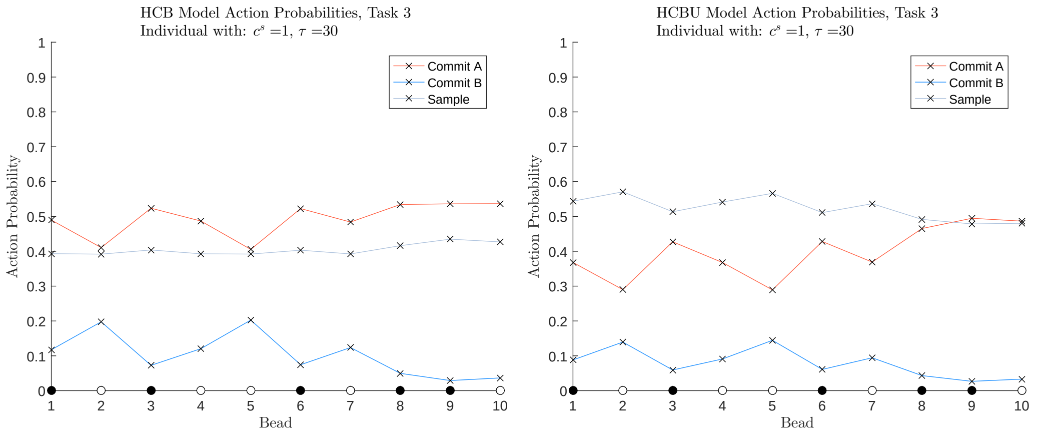

Figure 3.2: HCB vs HCBU model. The IBA with c^s would have chosen to sample until the 9th bead was drawn. There, he would have chosen jar A. In the HCBU model, the relativeness of the actions are preserved. If the IBA thinks that sampling is the best option, choosing jar A is the second best, and choosing jar B is the worst alternative, then the agent in the HCBU model will have largest probability of choosing to sample, second largest probability of choosing jar A, and smallest probability of choosing jar B. In the HCB model however, the noise may cause the probability of choosing jar A to move beyond the probability of sampling. So in the HCB model, large noise provoke faster decisions.

number of instances of choosing the wrong[2] jar, however, it can only be explained by a large τ. This model is the same as the one introduced by Moutoussis *et al.*.

3.3.2 The HCBU Model

In the HCBU model, the agent assumes that he is an IBA when he calculates the additional gain expected to be obtained when sampling another bead. He uses (3.6) when calculating (3.8), which does not involve τ. Therefore, two agents with different τ will obtain the exact same action values for sampling another bead. As τ increases, the probability of choosing each of the available actions get closer to each other as we can see from (3.10). This may cause the probability of sampling to decrease. However, in contrast to the agent in the HCB model, the probability of sampling does not decrease as a consequence of the action value for sampling decreasing. In this manner, the HCBU model removes some of the "overlap" between the parameters c^s and τ that exists in the HCB model.

[2] Wrong here means choosing a jar that is not the posteriorly more likely jar.

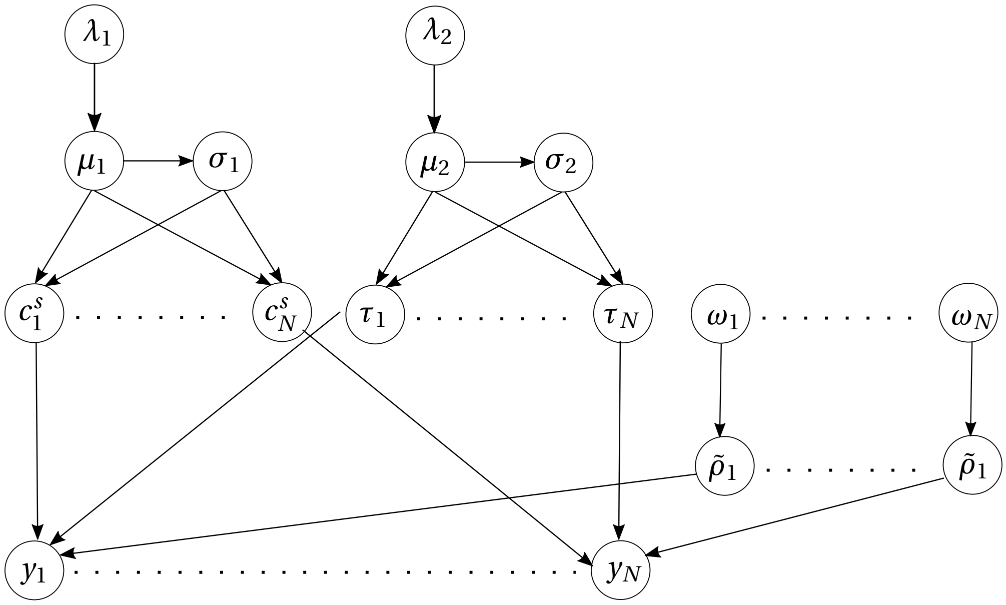

Figure 3.3: The structure of the HCBP and HCBUP models. Instead of considering the parameters α, β, η, and δ, we consider μ_1, σ_1, μ_2, and σ_2. The constants λ_1 and λ_2 are concerned with the hyper-prior that is used in the Bayesian analysis as discussed in section 4.2. The y's and $\tilde{\omega}$'s are observations, the λ's are chosen constants, and the rest of the parameters are latent variables.

3.3.3 Incorporating the Likelihood Ratings - The HCBP and HCBUP Models

The model(s) we have built so far tries to account for the participants' eagerness to make a decision as well as their randomness in choosing between options. Now, we would like to extend the model so that it also captures the participant's noise in estimating probabilities. We do this by incorporating the likelihood ratings, which the participants provide after each bead is drawn, in the model. Let us denote the likelihood ratings for jar A and jar B as L_A and L_B, respectively. These estimates are provided by the participants through sliders ranging from "Can't be this lake" to "Must be this lake". There are no constraints regarding the relationship between the likelihood ratings; for example, both ratings may be set to the value "Must be this lake". Therefore, it is not obvious how we should interpret these ratings. Some participants may assume that the sliders range from 0 to 1, and that they represent the posterior probability of the corresponding jar being the source of the beads. In this case, the likelihood ratings provided should sum to 1. In order for us to incorporating these ratings in the model, we need to transform them to something meaningful. We cannot simply let them range from 0 to 1 and say that they represent the posterior probability of each of the jars. This is because the ratings provided by the participants rarely sum to one, and this does not make sense theoretically. Instead, we define the posterior probability estimate for jar v to be

$$\tilde{p}_v = \frac{L_v}{\sum_{\tilde{v}\in\{A,B\}} L_{\tilde{v}}}, \quad v \in \{A,B\}. \tag{3.11}$$

We see that for the individuals that in fact interpreted the slider values as posterior probability estimates for jar v, we get $\tilde{p}_v = L_v$. Under this transformation, a person who rates the two jars as "Can't be this lake" and a person who rates the two jars as "Must be this lake" are both saying that there is a 50% chance of each of the jars being the source. The person that rates both jars as very unlikely may do so because he thinks that both jars are equally likely of being the source, but at the same time he thinks that the displayed beads are not very representative for any of the two jars. The person who rates both as very likely may do so because he thinks that they are equally likely, and that the displayed beads are very representative for both of the jars.

In contrast to the IBA, participants are not able to correctly calculate the posterior probability of a jar being the source of the beads. Participants' probability estimates are subject to noise. As noted in Chapter 2, we can add normally distributed noise to the log-odds transformation of the true probability of jar A. Let p_A be the true probability that jar A is the source of the beads in a given state, and let $\tilde{p}_A$ the individual's estimate of this probability. Then, we get

$$\log(\frac{\tilde{p}_A}{1-\tilde{p}_A}) = \log(\frac{p_A}{1-p_A}) + \omega \cdot U, \tag{3.12}$$

where $U \sim N(0,1)$. This is equivalent to

$$\tilde{p}_A = \frac{p_A}{p_A + (1-p_A)e^{-\omega\cdot U}}$$

The probability estimate of jar B being the source is $\tilde{p}_B = 1 - \tilde{p}_A$. The parameter ω says how much the probability estimate provided by a participant tends to deviate from the true probability. The IBA has $\omega = 0$, while an individual that has difficulties in estimating the posterior probability has a large ω. In the case where there are more than two jars, we can make a similar transformation by setting one of the jars as *baseline*, and use the fact that the probabilities for the jars sum to one. This is the same technique that is used in the multinomial logit model, and we refer to [6] for the technical details.

So how should we incorporate the noise in probability estimates in the model? As noted earlier, the process of calculating the action value of sampling another bead requires the agent to calculate the expected gain in all the future states that may be visited. When performing these calculations, the IBA utilizes the value of p_A for the current state and all the possible future states. Originally, we wanted to construct a model where there is noise every time the agent calculates a p_A. However, this requires us to integrate out all the unobserved U's, which turns out to be a non-trivial task. Instead, we simplify the model by letting the agent have noise only when calculating p_A for the *current* state that he finds himself within. These estimates are actually "observed" as $\tilde{p}_A$. We write observed in quotation mark because they are strictly speaking not observed; instead, L_A and L_B are observed, and these are utilized to find the individual's estimates of p_A. The structure is illustrated in Figure 3.3 Under this model, when an agent is in state $s_i = n_w$, he calculates the action value for choosing a jar $v \in \{A, B\}$ as

$$Q(d^v, i, n_w, \mathbb{S}) = (r + c_w) \cdot \tilde{p}_v - c_w, \tag{3.13}$$

where the $\tilde{p}_A$ corresponds to the estimate the participant provides in state $s_i = n_w$. This is a modified version of (3.3). Furthermore, an agent in state $s_i = n_w$ calculates the action value for sampling another bead as

$$Q(d^S, i, n_w, \mathbb{S}) = \sum_{v \in \{A,B\}} \tilde{p}_v \cdot h(i, n_w, v, d^S, \mathbb{S}), \tag{3.14}$$

which is (3.9) where $P(V = v|S_i = n_w)$ has been replaced by the subjective estimate $\tilde{p}_v$. This setting gives rise to two alternative models, as for the HCB and HCBU models. In the first model, the agents know that they have decision noise. We call this model HCBP, where the "P" is there to denote that the subjective probability estimates are taken into account. In the second model, the agents are unaware that they have decision noise. We refer to this model as the HCBUP model.

Chapter 4

Parameter Estimation

So far, we used the IBA as a reference and constructed four models that describe the behavior of human beings performing the beads task. Two of these models, the HCB and HCBU models, account for bias towards making a hasty conclusion (through c^s) and decision noise (through τ). The two other models, the HCBP and HCBUP models, additionally try to capture the noise in estimating probabilities (through ω). In this chapter, we describe how we estimate the parameters in these models. First, we only look at the HCB and HCBU models. We start off by looking at how the parameters can been obtained through maximum likelihood estimation (MLE). However, for reasons that we discuss later, we do not carry out the MLE estimation in practice. Then, we move on to Bayesian parameter estimation. In this setting, we view the model parameters as stochastic variables instead of fixed parameters. We make inference about the parameters based on the resulting posterior density, by calculating the marginal means and variances. As we cannot perform the integration of the posterior density analytically, we utilize the M-H algorithm as a tool for numerical integration. We round off this chapter by showing how we estimate the parameters in the HCBP and HCBUP models.

Before we look at the parameter estimation, we need to make a couple of modifications to the models so that they are compatible with the experimental data we posses. In the models we have looked at so far, the participants must make a decision at some point in the task. As noted earlier, however, the participants in the experiment are not required to make a commitment to a jar during the task. They can sample through all the n beads and say that they are still not sure regarding the source of the beads when the n-th bead is drawn. We handle this by imagining that when the participants calculate the action values, they assume that there is no limit on the number of beads they can see before making a decision. In the existing models, we can let the individuals assume that they can see n_{assumed} number of beads instead of only n number of beads in the task. As n_{assumed} goes to infinity, the individuals assume that there is no limit on the number of beads they are allowed to see before making a decision, and they act accordingly. In theory, we can choose n_{assumed} as large as we want in order to approximate that there is no limit to the number of beads that can be seen. In practice, however, the parameter estimation becomes very computationally expensive as n_{assumed} increases. Therefore, we need to give it moderately large value. In the following, the action values for sampling, $Q(d^S,\cdot,\cdot,\cdot,\cdot)$, are calculated as if n_{assumed} number of beads can be drawn instead of only n. We also need to make a modification to the data set; some participants provide probability estimates p_v that are zero or one. Our model for how these are generated does not allow for these to actually reach zero or 1, since the logit function only approach 0 and 1 asymptotically. We handle this by adding 10^4 to the instances that are 0, and subtracting this value to instances that are 1.

4.1 Maximum Likelihood Estimation

In this section, we derive the likelihood function of the hyper-parameters in the HCB model. Let us first look at the likelihood function for the parameters of a single individual under these two models. We use the notation introduced earlier, but we add an extra t to the subscript to denote which task we are considering, ranging from 1 to N_{tasks} . That is, we denote x_{ti} as the color of bead i in task t and $\mathbf{x}_t = \{x_{t1},...,x_{tn}\}$ as the entire displayed sequence in task t. Additionally, we introduce $x = \{\mathbf{x}_t\}$ as the collection of all the displayed sequences that are given to the agents. Also, we let $r = \{(r_{tA}, r_{tB})\}$ contain the ratio of white beads in each of the jars in each of the trials. Now, we let the subscript $k \in \{1,...,N_{\text{ind}}\}$ denote the agent in question. Then, we denote d_{tik} as the action taken by agent k in trial t after i beads have been drawn. Furthermore, we let $y_k = \{(m_{tk}, d_{tk})\}$ denote the responses of individual k on each of the tasks, containing m_{tk} which is the number of beads displayed before a decision was made in task t, and d_{tk} which says which jar that was chosen eventually in task t. If the individual did not commit to a jar in the task, we define m_{tk} to take the value $n+1$ for the sake of notation and d_{tk} to take the value 0. Finally, let $y = \{y_k : k = 1, ..., N_{\text{ind}}\}$ be the collection of the responses from all the individuals. Then, the likelihood function for the micro-parameters of one single individual can be written as

$$\begin{aligned}
p(y_k|c_k^s, \tau_k, x, r, \mathbb{S}) &= \prod_{t=1}^{N_{\text{tasks}}} \big(P(D_{tm_{tk}} = d_{tk}|x, r, \mathbb{S})\big)^{I[m_{tk}\neq n+1]} \prod_{i=1}^{m_{tk}-1} P(D_{tik} = d^S|x, r, \mathbb{S}) \\
&= \prod_{t=1}^{N_{\text{tasks}}} \left(w(d_{tk}, m_{tk}, \sum_{s=1}^{m_{tk}} x_{ts}, r, \mathbb{S}) \right)^{I[m_{tk}\neq n+1]} \prod_{i=1}^{m_{tk}-1} w(d^S, i, \sum_{s=1}^{i} x_{ts}, r, \mathbb{S}) \\
&= \prod_{t=1}^{N_{\text{tasks}}} \left(\frac{e^{Q(d_{tk}, m_{tk}, \sum_{s=1}^{m_{tk}} x_{ts}, \mathbb{S})/\tau_k}}{\sum_{d\in\mathbb{D}_{tm_{tk}}} e^{Q(d, m_{tk}, \sum_{s=1}^{m_{tk}} x_{ts}, \mathbb{S})/\tau_k}} \right)^{I[m_{tk}\neq n+1]} \prod_{i=1}^{m_{tk}-1} \frac{e^{Q(d^S, i, \sum_{s=1}^{i} x_{ts}, \mathbb{S})/\tau_k}}{\sum_{d\in\mathbb{D}_{ti}} e^{Q(d, i, \sum_{s=1}^{i} x_{ts}, c_k^S, \mathbb{S})/\tau_k}},
\end{aligned}$$

where we have used the indicator function $I[\cdot]$ in order to remove the first term from the product if the participant does not commit to any jar in the task. In the hierarchical model, the contribution to the likelihood function from a single individual becomes

$$\begin{aligned}
L_k^{HCB}(\alpha, \beta, \eta, \delta|y_k, x, r) &= p(y_k|\alpha, \beta, \eta, \delta, x, r) \\
&\overset{LTP}{=} \int_{c^S} p(c^S|\alpha, \beta, \eta, \delta, y_k, x, r) \cdot p(y_k|c^S, \alpha, \beta, \eta, \delta, x, r) dc^S \\
&= \int_{c^S} p(c^S|\alpha, \beta) \cdot p(y_k|c^S, \eta, \delta, x, r) dc^S \\
&\overset{LTP}{=} \int_{c^S} p(c^S|\alpha, \beta) \int_{\tau} p(\tau|\eta, \delta) p(y_k|c^S, \tau, x, r) d\tau dc^S \\
&= \int_{c^S} \int_{\tau} p(c^S|\alpha, \beta) p(\tau|\eta, \delta) p(y_k|c^S, \tau, x, r) d\tau dc^S
\end{aligned}$$

where $p(c^S|\alpha, \beta)$ and $p(\tau|\eta, \delta)$ are the probability density functions of the gamma distribution, respectively. The full likelihood function is the product of the likelihood functions for each individual, and it can be evaluated by using numerical integration to evaluate the double integrals. The MLE estimates can be found by maximizing this function with respect to α, β, η, and δ through numerical maximization. However, as the integrands are computationally expensive to evaluate, it turns out that maximizing this function numerically takes is to time consuming. Instead, we would like to impose an estimation procedure that does not require integrating out the latent individual-specific parameters c^s and τ. Moutoussis *et al.* rely on the Expectation-maximization algorithm to obtain the MLE parameters. We, on the other hand, turn to Bayesian estimation and look at the posterior distribution of all the model parameters.

4.2 Bayesian Parameter Estimation

In the Bayesian setting, we treat the model parameters as random variables. We set a prior on the model parameters and from this prior we obtain their posterior distribution. Then, we can make inference about the parameters by calculating the mean and variance of the posterior distribution. Before performing the Bayesian analysis, we make a convenient re-parametrization of the model: instead of dealing with α and β as the parameters of the gamma distribution, we use the mean and standard deviation (SD) of the gamma distribution, i.e. $\mu_1 = \alpha\beta$ and $\sigma_1 = \sqrt{\alpha}\beta$. Similarly, we work with $\mu_2 = \eta\delta$ and $\sigma_2 = \sqrt{\eta}\delta$ instead of η and δ. These are more intuitive parameters to deal with because they represent the mean and SD of the sampling costs and the mean and SD of the noise parameters and we have some understanding of what these parameters mean.

4.2.1 Choice of Hyperprior

Let us treat the hyper-parameters μ_1, σ_1, μ_2 and σ_2 as random variables. We need to choose a prior for the hyper-parameters, i.e. a hyperprior. There is no obvious choice for this prior. We could use a diffuse hyperprior that gives vague information about the parameters. In this case, the posterior is mainly driven by the observations. However, as Moutoussis *et al.* has performed a similar analysis before us, we possess some prior knowledge about the parameters. We may use this information when constructing our hyperprior. In particular, we may choose a hyperprior with a mean that has a value close to the value of the MLE parameters obtained by Moutoussis *et al.* Furthermore, there is no reason to believe that there is a dependency between the governing parameters for the sampling cost and the governing parameters for the noise parameter, so we choose a prior such that (μ_1, σ_1) and (μ_2, σ_2) are independent. Additionally, we note that the sampling costs and the noise parameters are non-negative. With this in mind, we choose that the mean of the sampling cost (μ_1) is exponentially distributed with a mean equal to some chosen constant λ_1. That is,

$$\mu_1 \sim \text{Exp}\left(\frac{1}{\lambda_1}\right), \tag{4.1}$$

so that the parameter is positive with an expected value $\frac{1}{\lambda_1}$. We use results obtained Moutoussis *et al.* to set a value for λ_1. Also, we would like the SD of the sampling cost (σ_1) to be of the same magnitude as the mean of the sampling cost (μ_1). Therefore, we choose

$$\sigma_1|\mu_1 \sim \text{Exp}\left(\frac{1}{k\mu_1}\right), \tag{4.2}$$

so that $E[\sigma_1|\mu_1] = k\mu_1$ where we can set a constant $0 < k < 1$. We use the same reasoning to obtain the hyper-prior related to μ_2 and σ_2, and we get

$$\mu_2 \sim \text{Exp}\left(\frac{1}{\lambda_2}\right) \tag{4.3}$$

and

$$\sigma_2|\mu_2 \sim \text{Exp}\left(\frac{1}{k\mu_2}\right). \tag{4.4}$$

The equations (4.1) to (4.4) give us the prior density function of the hyperparameters, namely

$$p(\mu_1, \sigma_1, \mu_2, \sigma_2) = p(\mu_1)p(\sigma_1|\mu_1)p(\mu_2)p(\sigma_2|\mu_2), \tag{4.5}$$

where all the densities are the PDF of the exponential distribution. Now, let us write out the expression for the joint posterior density of all the parameters in the model, $p(\boldsymbol{\theta}|y, x, r)$, where $\boldsymbol{\theta} = (\mu_1, \sigma_1, \mu_2, \sigma_2, c_1^s, ..., c_{N_{\text{ind}}}^s, \tau_1, ..., \tau_{N_{\text{ind}}})$ is the vector containing all the model parameters. By

Bayes Rule (BR) and by the construction of the model, we have

$$\begin{aligned}
p(\boldsymbol{\theta}|y,x,r) &\overset{BR}{=} \frac{p(\boldsymbol{\theta}|x,r)p(y|\boldsymbol{\theta},x,r)}{p(y|x,r)} \\
&= \frac{p(\boldsymbol{\theta})p(y|c_1^s,..,c_{N_{\text{ind}}}^s,\tau_1,...,\tau_{N_{\text{ind}}},x,r)}{p(y|x,r)} \\
&= \frac{p(\mu_1,\sigma_1,\mu_2,\sigma_2)p(c_1^s,...,c_{N_{\text{ind}}}^s|\mu_1,\sigma_1)p(\tau_1,...,\tau_{N_{\text{ind}}}|\mu_2,\tau_2)p(y|c_1^s,..,c_{N_{\text{ind}}}^s,\tau_1,...,\tau_{N_{\text{ind}}},x,r)}{p(y|x,r)} \\
&= \frac{p(\mu_1)p(\sigma_1|\mu_1)p(\mu_2)p(\sigma_2|\mu_2)\prod_{i=1}^{N_{\text{ind}}}p(c_i^s|\mu_1,\sigma_1)p(\tau_i|\mu_2,\sigma_2)p(y_i|c_i^s,\tau_i,x,r)}{p(y|x,r)} \\
&\propto p(\mu_1)p(\sigma_1|\mu_1)p(\mu_2)p(\sigma_2|\mu_2)\prod_{i=1}^{N_{\text{ind}}}p(c_i^s|\mu_1,\sigma_1)p(\tau_i|\mu_2,\sigma_2)p(y_i|c_i^s,\tau_i,x,r), \qquad (4.6)
\end{aligned}$$

where all the densities involved already have been presented. This density function provide us with posterior information about the model parameters. Ideally, we would calculate sample mean and variance analytically by integrating this function. However, analytical solutions cannot be obtained in this case. Instead, we choose to utilize the M-H algorithm in order sample from the distribution. Through this sample, we empirically obtain information about the posterior distribution of the parameters in the model. For example, the mean of the distribution is estimated by calculating the sample average. We have omitted the normalizing constant in the last line in (4.6) since the M-H algorithm only needs the ratio of the posterior densities, causing the normalizing constant to cancel.

4.2.2 Choice of Proposal Distribution for M-H

We use the H-M algorithm to sample from the posterior in (4.6). As noted in Section 2.2, the algorithm requires us to choose a proposal distribution, which together with the acceptance probability generates the next state of the Markov chain. Though this function can be chosen arbitrarily, the choice may strongly affects the rate of convergence and the mixing of the chain. We recall that rate of convergence means how many iterations it takes for the chain to reach the stationary distribution from an arbitrary starting point. The mixing concerns how quickly the chain explores the support of the target distribution and how correlated the samples produces by the chain are. Both the rate of convergence and the mixing are influenced by the *spread* of the proposal distribution. Therefore, we should construct a proposal distribution for which the spread can be tuned. A way of checking whether the spread of the proposal density seems to be appropriate is by calculating the acceptance rate of the proposals. We bear this in mind when selecting a proposal distribution for our implementation.

As we would like to sample from the posterior density of $\boldsymbol{\theta}$, we set $\pi(\boldsymbol{\theta}) = p(\boldsymbol{\theta}|y,x,r)$ in the H-M algorithm described in Section 2.2. We use component-wise updating. This means that only one of the components of the parameter vector $\boldsymbol{\theta}$ may change value at each iteration while the other components remain as they were in the previous time step. This is convenient because it allows us to work with univariate distributions. Since all the parameters in the parameter vector by definition are non-negative, we would like to have proposal distributions that only sample positive candidates; a candidate with negative value is guaranteed to be rejected as the posterior density is zero, which in turn causes the acceptance probability to be zero. Letting θ_k be the k-th parameter in the parameter vector and $\tilde{\theta}_k$ be the k-th element of the candidate vector (with the rest of the elements in this candidate vector being equal to the the corresponding element in $\boldsymbol{\theta}^{(t)}$), we use a uniform proposal distributions that covers the current point,

$$\tilde{\theta}_k|\theta_k^{(t)} \sim \text{Unif}\left(\frac{\theta_k^{(t)}}{1+\epsilon_k}, \theta_k^{(t)}(1+\epsilon_k)\right), \quad \epsilon_k > 0, \quad k = 1,..,2\cdot N_{\text{ind}}+4. \tag{4.7}$$

Thus, we have the proposal density functions

$$q_k(\tilde{\boldsymbol{\theta}}|\boldsymbol{\theta}^{(t)}) = \begin{cases} \frac{1}{\left(1+\epsilon_k-\frac{1}{1+\epsilon_k}\right)\theta_k^{(t)}} \text{ if } \frac{1}{1+\epsilon_k}\theta_k^{(t)} < \tilde{\theta}_k < (1+\epsilon_k)\theta_k^{(t)}, \\ 0 \text{ otherwise.} \end{cases} \tag{4.8}$$

In other words, the candidate parameters are sampled uniformly within the interval $(\frac{\theta_k^{(t)}}{1+\epsilon_k}, \theta_k^{(t)}(1+\epsilon_k))$, which covers the current parameter value, $\theta_k^{(t)}$. These proposals will 1) ensure that the candidate parameters always are positive, and 2) allow for adjusting the spread through the tuning parameters ϵ_k. As ϵ_k increases, the spread/variance of the proposal gets larger since the support of the uniform proposal increases.

We proceed by writing out the Metropolis-Hastings ratio we get when using the proposal distributions given in (4.7). By inserting the posterior density given in (4.6), we get

$$
\begin{aligned}
R_k(\tilde{\boldsymbol{\theta}}|\boldsymbol{\theta}) &= \frac{p(\tilde{\boldsymbol{\theta}}|y,x,r,\mathbb{S})\,q_k(\boldsymbol{\theta}|\tilde{\boldsymbol{\theta}})}{p(\boldsymbol{\theta}|y,x,r,\mathbb{S})\,q_k(\tilde{\boldsymbol{\theta}}|\boldsymbol{\theta})} \\
&= \frac{p(\tilde{\mu}_1)p(\tilde{\sigma}_1|\tilde{\mu}_1)p(\tilde{\mu}_2)p(\tilde{\sigma}_2|\tilde{\mu}_2)\prod_{i=1}^{N_{\text{ind}}} p(\tilde{c}_i^s|\tilde{\mu}_1,\tilde{\sigma}_1)p(\tilde{\tau}_i|\tilde{\mu}_2,\tilde{\sigma}_2)p(y_i|\tilde{c}_i^s,\tilde{\tau}_i,x,r,\mathbb{S})}{p(\mu_1)p(\sigma_1|\mu_1)p(\mu_2)p(\sigma_2|\mu_2)\prod_{i=1}^{N_{\text{ind}}} p(c_i^s|\mu_1,\sigma_1)p(\tau_i|\mu_2,\sigma_2)p(y_i|c_i^s,\tau_i,x,r,\mathbb{S})} \cdot \frac{\theta_k}{\tilde{\theta}_k}.
\end{aligned}
$$

Since only parameter number k is sampled in each iteration, all the terms that do not involve this parameter cancel. Therefore, if $k = 1$ then $\theta_1 = \mu_1$ is the parameter that is sampled, and all the terms that do not involve this parameter vanish. Thus, we get

$$
R_1(\tilde{\boldsymbol{\theta}}|\boldsymbol{\theta}) = \frac{p(\tilde{\mu}_1)}{p(\mu_1)}\frac{p(\sigma_1|\tilde{\mu}_1)}{p(\sigma_1|\mu_1)}\prod_{i=1}^{N_{\text{ind}}}\frac{p(c_i^s|\tilde{\mu}_1,\sigma_1)}{p(c_i^s|\mu_1,\sigma_1)} \cdot \frac{\mu_1}{\tilde{\mu}_1}
$$

When $k = 2$, the parameter $\theta_2 = \sigma_1$, is sampled and the M-H ratio is

$$
R_2(\tilde{\boldsymbol{\theta}}|\boldsymbol{\theta}) = \frac{p(\tilde{\sigma}_1|\mu_1)}{p(\sigma_1|\mu_1)}\prod_{i=1}^{N_{\text{ind}}}\frac{p(c_i^s|\mu_1,\tilde{\sigma}_1)}{p(c_i^s|\mu_1,\sigma_1)} \cdot \frac{\sigma_1}{\tilde{\sigma}_1}.
$$

For $k = 3$, the parameter $\theta_3 = \mu_2$ is sampled and we get

$$
R_3(\tilde{\boldsymbol{\theta}}|\boldsymbol{\theta}) = \frac{p(\tilde{\mu}_2)}{p(\mu_2)}\frac{p(\sigma_2|\tilde{\mu}_2)}{p(\sigma_2|\mu_2)}\prod_{i=1}^{N_{\text{ind}}}\frac{p(\tau_i|\tilde{\mu}_2,\sigma_2)}{p(\tau_i|\mu_2,\sigma_2)} \cdot \frac{\mu_2}{\tilde{\mu}_2}.
$$

For $k = 4$, the parameter $\theta_4 = \sigma_2$ is sampled, and we get

$$
R_4(\tilde{\boldsymbol{\theta}}|\boldsymbol{\theta}) = \frac{p(\tilde{\sigma}_2|\mu_2)}{p(\sigma_2|\mu_2)}\prod_{i=1}^{N_{\text{ind}}}\frac{p(\tau_i|\mu_2,\tilde{\sigma}_2)}{p(\tau_i|\mu_2,\sigma_2)} \cdot \frac{\sigma_2}{\tilde{\sigma}_2}.
$$

These four first elements are the hyper-parameters. For $k \in \{5,\ldots,4+N_{\text{ind}}\}$ one of the sampling cost micro-parameters is sampled. So $\theta_k = c_i^s$, where $i = k-4$, and we get

$$
R_k(\tilde{\boldsymbol{\theta}}|\boldsymbol{\theta}) = \frac{p(\tilde{c}_i^s|\mu_1,\sigma_1)}{p(c_i^s|\mu_1,\sigma_1)}\frac{p(y_i|\tilde{c}_i^s,\tau_i,x,r,\mathbb{S})}{p(y_i|c_i^s,\tau_i,x,r,\mathbb{S})} \cdot \frac{c_i^s}{\tilde{c}_i^s}.
$$

For $k \in \{5+N_{\text{ind}}, ..., 2N_{\text{ind}}\}$, one of the noise micro-parameters are sampled. So $\theta_k = \tau_i$ where $i = k-4-N_{\text{ind}}$, and we get

$$R_k(\tilde{\boldsymbol{\theta}}|\boldsymbol{\theta}) = \frac{p(\tilde{\tau}_i|\mu_2,\sigma_2)}{p(\tau_i|\mu_2,\sigma_2)} \frac{p(y_i|c_i^s,\tilde{\tau}_i,x,r,\mathbb{S})}{p(y_i|c_i^s,\tau_i,x,r,\mathbb{S})} \cdot \frac{\tau_i}{\tilde{\tau}_i},$$

where $i = k-4-N_{\text{ind}}$. The M-H ratios R_1, R_2, R_3 and R_4 are relatively computationally inexpensive to calculate as they only involve the gamma and exponential PDF's. For $k > 4$, however, R_k involves calculating $p(y_i|c_i^s,\tau_i,x,r,\mathbb{S})$ which requires some computation time. When implementing the M-H algorithm, we use two 'tricks' to make the algorithm as efficient as possible. First of all, we make sure that $\theta_1,\theta_2,\theta_3$ and θ_4 are sampled more frequently than θ_k, $k > 4$. Specifically, we have chosen that 50% of the iterations are designated the first four parameters. Secondly, we continually store the likelihood values $p(y_i|c_i^s,\tau_i,x,r,\mathbb{S})$ for each of the individuals. Then, we only have to evaluate the computationally expensive likelihood value once instead of twice every time we calculate the M-H ratio for $k > 4$. We also note that we always work on logarithmic scale when calculating the probabilities, ratio of probabilities and product of probabilities in the implementations. We do this to avoid problems caused by numerical instability. In the next chapter, we perform a simulation study and sensitivity analysis to see how well the algorithm works.

4.3 Parameter Estimation in the HCBP and HCBUP Models

As we recall from section 3.3.3, the HCBP model adds yet another individual-specific parameter to each individual. In order for the HCBP model to be fully Bayesian, we need to put a prior [1] on the parameter vector $\boldsymbol{\theta}^* = (\omega_1, ..., \omega_{N_{\text{ind}}})$, and write out the posterior density of the model parameters. As before, we let y be the collection of all the observations that have to do with the DTD responses, and $\boldsymbol{\theta} = (\mu_1,\sigma_1,\mu_2,\sigma_2,c_1^s,...,c_{N_{\text{ind}}}^s,\tau_1,...,\tau_{N_{\text{ind}}})$. Furthermore, we let $\tilde{\rho}_k$ be all the probability estimates provided by participant k and $\tilde{\rho}$ be the collection of all the participants'

[1] Alternatively, we could build an hierarchical structure by having a prior with unknown parameters and putting a hyperprior with known parameters on top.

estimates. Then, the posterior density in this model can be written as

$$\begin{aligned} p(\boldsymbol{\theta},\boldsymbol{\theta}^*|y,\tilde{\rho},x,r) &\propto p(y,\tilde{\rho}|\boldsymbol{\theta},\boldsymbol{\theta}^*,x,r)p(\boldsymbol{\theta},\boldsymbol{\theta}^*) \\ &= p(\tilde{\rho}|\boldsymbol{\theta},\boldsymbol{\theta}^*,x,r)p(y|\boldsymbol{\theta},\boldsymbol{\theta}^*,\tilde{\rho},x,r)p(\boldsymbol{\theta},\boldsymbol{\theta}^*) \\ &= p(\tilde{\rho}|\boldsymbol{\theta}^*,x,r)p(y|\boldsymbol{\theta},\tilde{\rho},x,r)p(\boldsymbol{\theta})p(\boldsymbol{\theta}^*) \\ &= p(\tilde{\rho}|\boldsymbol{\theta}^*,x,r)p(\boldsymbol{\theta}^*)\cdot p(y|\boldsymbol{\theta},\tilde{\rho},x,r)p(\boldsymbol{\theta}). \end{aligned}$$

As the posterior density can be written as a product of one function that involves $\boldsymbol{\theta}^*$ and one that involves $\boldsymbol{\theta}$, these parameter vectors are posteriorly independent. As a consequence, we can make inference on these parameter vectors separately. The estimate of $\boldsymbol{\theta}^*$ obtain the exact same value in the HCBP and HCBUP models. We perform a Bayesian analysis on $\boldsymbol{\theta}$, using the same prior as we did for the HCB and HCBU models in section 4.2.1. To access information about the posterior density, we utilize the M-H algorithm with the same proposal distribution as for the HCB and HCBU models. To infer about $\boldsymbol{\theta}^*$, on the other hand, we use MLE. We could have estimated this vector as Bayesians, but we find MLE more convenient because the estimates have a simple analytical solution in this case. Next, we write out the expression for the MLE estimate of ω^2 for a single individual. Let $\rho_{A,l}$ be the probability of jar A being the source of the beads, where l ranges over all the N states where the participant provides a probability estimate. Similarly, let $\tilde{\rho}_{A,l}$ be the probability estimate that the particular individual actually provides. Then, define $z_l = \log(\frac{\rho_{A,l}}{1-\rho_{A,l}})$ and $\tilde{z}_l = \log(\frac{\tilde{\rho}_{A,l}}{1-\tilde{\rho}_{A,l}})$, which are the probabilities mapped to the real line. In our model, we have $\tilde{z}_l \sim N(z_l,\omega^2)$. By writing out the log-likelihood function, differentiating with respect to ω^2 and setting the expression equal to zero, we obtain

$$\hat{\omega}^2 = \frac{1}{N}\sum_{l=1}^{N}(\tilde{z}_l - z_l)^2. \tag{4.9}$$

This expression gives us the estimates of ω for each of the individuals in the data set.

In this chapter, we have proposed a procedure of estimating the parameters in the four models. Before applying these estimation procedures on the actual data set, we would like to see how well they work. In order to do so, we perform a simulation study. We implicitly also get to check whether there seems to be any programming errors. In the next chapter, we carry out the simulation study.

Chapter 5

Simulation Study

In this chapter, we simulate some experiment responses under the model(s) built in Chapter 3 and use our implementation of the H-M algorithm to sample from the resulting posterior distribution. There are two main purposes of performing this analysis. First of all, we would like to see whether our choice of proposal distribution in the algorithm works well. This involves investigating the convergence and the mixing of the Markov chain. Secondly, we aim to figure out how accurately we are able to retain the parameters in the model when using data from a reasonable number of participants and tasks per participant. This involves checking how sensitive the algorithm is to the choice of hyperprior. However, we implicitly also get to confirm that the implementation is made correctly. If there are any major programming errors, these should be detected during this analysis. We have carried out a simulation study on all the four models, but as the results are fairly similar, we only present the simulation study on the HCB model in this report.

5.1 M-H Simulations Under Different Scenarios

We run a total of seven simulations in this study. Each simulation consists of two steps. The first step is response generation. The responses are generated by first choosing some hyperparameters, then sampling micro-parameters for a chosen number of individuals, and finally simulating responses to some chosen tasks for each of these individuals. The second step is parameter estimation. Here, we specify some parameters for the hyperprior, and run H-M algorithm with some chosen tuning parameters on the responses that were generated.

The setup for each of the simulations is summarized in table 5.1. They all have in common that $n = n_{\text{assumed}} = 10$. Hence, ten beads can be drawn, and the participants know that they must make a commitment to a jar at some point in the sequence of ten beads. As we recall from Section 4.2.2, the proposal distribution is tuned by the ϵ_k parameters, each of these specifying the possible step size when there is a proposal for component k of $\boldsymbol{\theta}^{(t)}$. It is not necessary to tune the proposal distributions for the c^s parameters individually, so we set $\epsilon_5 = ... = \epsilon_{4+N_{\text{ind}}}$. The same goes for the τ parameters, so we give all of them the same tuning by setting $\epsilon_{4+N_{\text{ind}}} = ... = \epsilon_{4+2\cdot N_{\text{ind}}}$. Now, we define the vector $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \epsilon_2, \epsilon_4, \epsilon_4, \epsilon_5, \epsilon_{5+N_{\text{ind}}})$, which contains all the information about the tuning of the proposal distribution.

Each of the simulations serves to illustrate some point, either by itself or when compared to another simulation. From one simulation to another, we modify only one component in order to see the effect of changing that particular component. Before we go into the details of each simulation, let us give a brief overview of what points we are trying to bring forth through these simulations. Simulation 1 and simulation 2 aim to illustrate the effect of tuning the proposal distribution. Simulation 1 and simulation 3 are designed to show the impact of including more individuals in the data set. Simulation number 1 and 4 show the consequence of giving more tasks per individual. Simulation number 1 and 5 aim to illustrate how sensitive the parameter estimation is to the choice of hyperprior parameters λ_1 and λ_2.

In the first scenario, we have chosen some moderately [1] large values for hyperparameters, namely $\mu_1 = 2, \sigma_1 = 1, \mu_2 = 5$, and $\sigma_2 = 3$. We sample 100 individuals based on these hyperparameters, and let them respond to the four tasks given in Table 1.1. The "average individual" generated under this scenario has parameters $c^s = \mu_1 = 2$ and $\tau = \mu_2 = 5$. With these parameters, the probability of committing to jar A, jar B, and choosing to continue sampling are respectively 14%, 0.0001%, and 86% after one beads is drawn in task 1. In Figure 5.1, we present the responses/observations that were produced. For the parameter estimation, we choose the values

[1] Recall that we have chosen $r = 0$ and $c^w 100$ as references for the model.

Table 5.1: The scenarios in the simulation study. The symbol '*' Means that $\boldsymbol{\epsilon} = (1,1,1,1,1,1)$, and '**' means that $\boldsymbol{\epsilon} == (0.5,0.5,0.5,0.5,2,2)$

	Data Generation					Experiment		Hyperprior			Metropolis-Hasings		
SIM #	μ_1	σ_1	μ_2	σ_2	N_{ind}	n_{trials}	$n = n_{\text{assumed}}$	λ_1	λ_2	k	$\boldsymbol{\epsilon}$	$\boldsymbol{\theta}^{(0)}$	N_{it}
1	2	1	5	3	100	4	10	0.25	0.25	0.5	*	10	$1.0 \cdot 10^6$
2	2	1	5	3	100	4	10	0.25	0.25	0.5	**	10	$1.0 \cdot 10^6$
3	2	1	5	3	200	4	10	0.25	0.25	0.5	*	10	$1.2 \cdot 10^6$
4	2	1	5	3	100	8	10	0.25	0.25	0.5	*	10	$1.0 \cdot 10^6$
5	2	1	5	3	100	4	10	0.1	0.1	0.5	*	10	$1.0 \cdot 10^6$

$\lambda_1 = \lambda_2 = 0.25$ in the hyperprior. Therefore, hyperprior says that $E[\mu_1] = E[\mu_2] = 1/0.25 = 4$, which is reasonably close to the true values. To study how fast the M-H algorithm converges, we choose to let all the components of $\boldsymbol{\theta}^{(0)}$ have the value 10 so that they are reasonably far away from the values that were used to generate the responses. Let us take a look at the M-H simulation for this scenario. In Figure 5.2, we show trace plots of the Markov chain for some selected parameters. One the first row, we have trace plots of the hyperparameters governing the sampling costs (namely μ_1 and σ_1) as well as the trace plot of an individual-specific sampling cost parameter (c_1^s). Similarly, on the second row, we have the trace plots of μ_2, σ_2, and τ_1. We ran $1 \cdot 10^6$ iterations of the M-H algorithm. From these trace plots, it seems that it takes a while before convergence is reached. The chain makes some large leaps in the first iterations, and then it gradually moves towards a state where the parameters have values that are close to those used for response generation. It looks like about $5 \cdot 10^4$ iterations are needed before all the parameters seem to fluctuate randomly about some value. These plots also give us an indication of bad mixing; we can see that the samples fluctuate about some mean in a cyclic pattern with low frequency. In other words, the samples seem to have a large auto-correlation. This is particularly seen for the hyper-parameters. As we recall, we have decided that as much as 50% of the iterations of the M-H algorithm should perturb one of the hyperparameters components of $\boldsymbol{\theta}$. Lowering this percentage may decrease the auto-correlation of the hyper-parameters.

In Figure 5.3, we have plotted the acceptance rate for each parameters. The acceptance rate for a parameter is the proportion of times the candidate was accepted when there was a proposal for that particular parameter. As we recall, the rule of thumb is that acceptance rates should lie

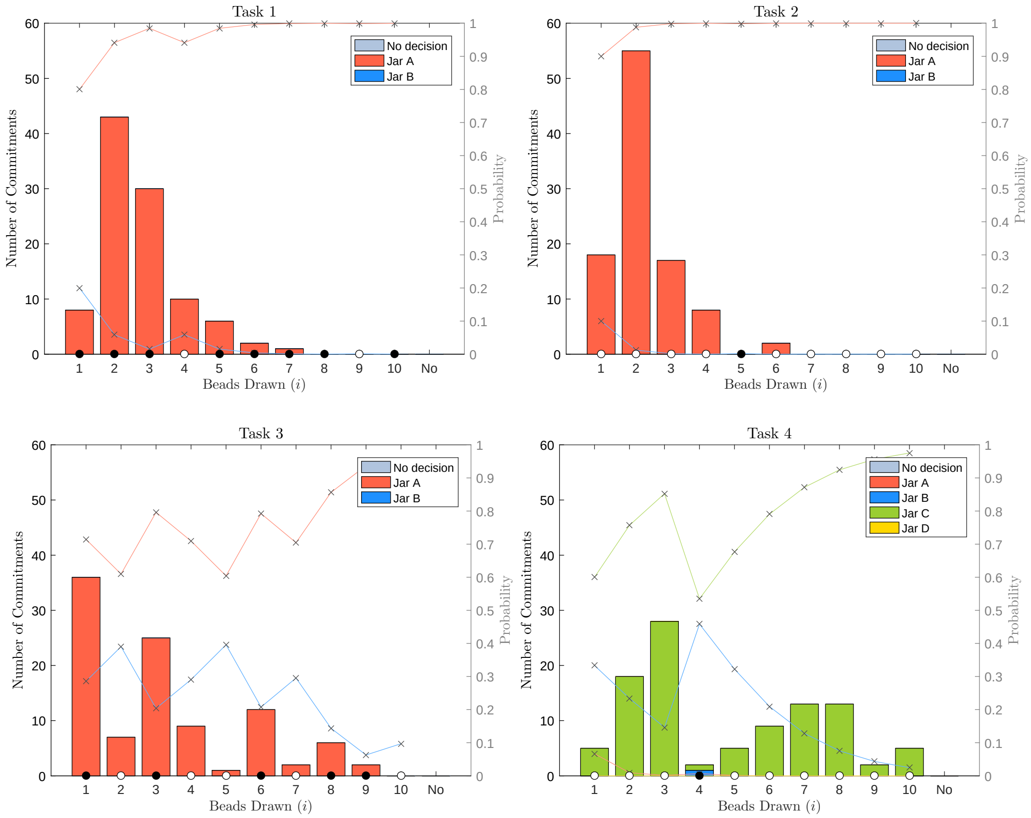

Figure 5.1: SIM1. The responses in simulation number 1. The plot shows how many individuals that responded after the first, second, etc bead was shown. There were in total 100 individuas in the generated data set. The true probability of each of the jars being the the source is displayed on the right y-axis. Table 1.1 shows the ratios of black, white, and red beads in each of these tasks. Since the participants had to make a commitment during each task, the 'No Commitment' bar is empty.

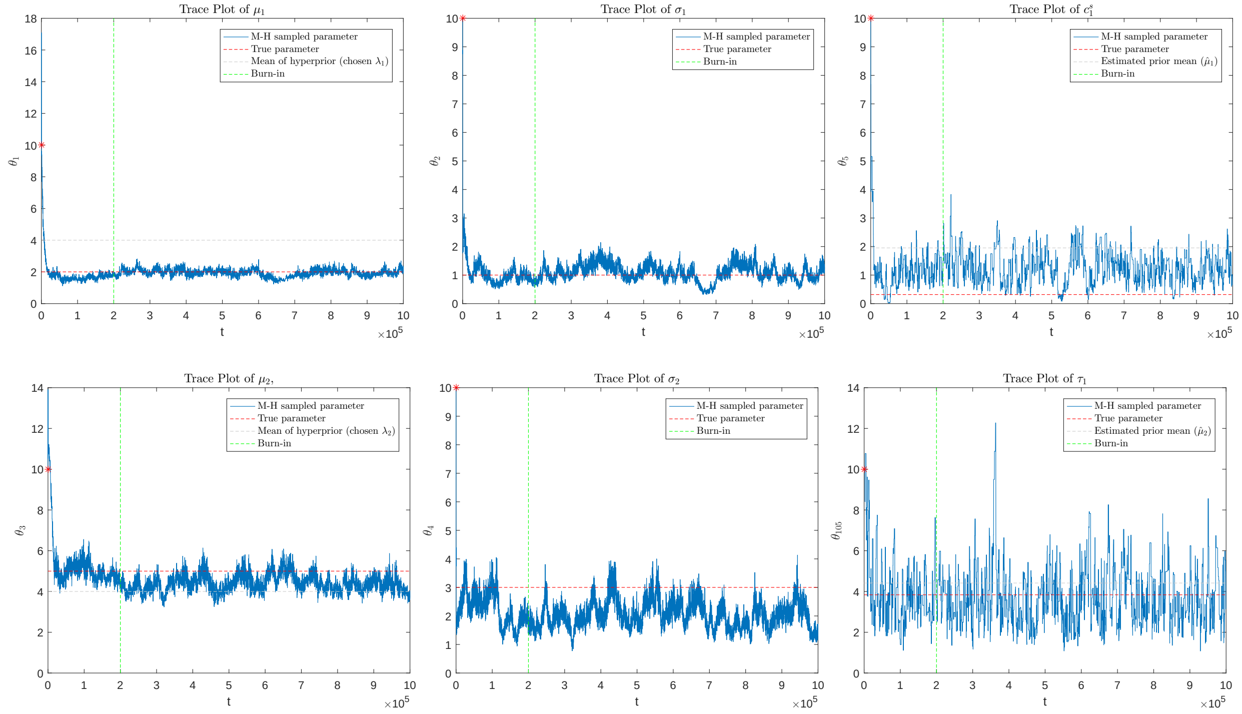

Figure 5.2: SIM1. Trace plots for some of the parameters in simulation study number 1. In the first row, the trace plots of the hyperparameters of c^s and one of the c^s parameters are plotted. The plots on the second row show the traces of the hyperparameters of τ as well as one of the τ parameters. We have chosen a burn-in period of $2 \cdot 10^5$ iterations.

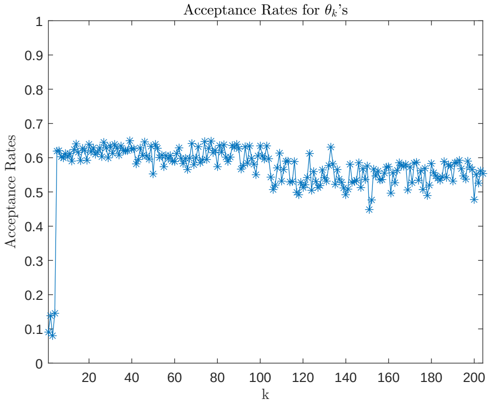

Figure 5.3: SIM1. Acceptance rates for the parameters in simulation study number 1.

between 20% and 50%. In this simulation, the acceptance rates are about 10% for the hyperparameters and 50% for the individual-specific parameters. Therefore, tuning the proposal by decreasing ϵ's for the hyperparameter proposals and increasing them for the proposals of c^s's and τ's should be considered.

Yet another way to monitor the Markov chain is by checking the correlation between the parameters in $\boldsymbol{\theta}^{(t)}$. The correlation matrix with the correlation between all pairs of parameters is visualized in the left panel in Figure 5.4. In the right panel of this figure, we can see the correlation between the hyperparameters only. These plots show that there is some positive correlation between μ_1 and σ_1, and between μ_2 and σ_2. These sample correlations are 0.64 and 0.55, respectively. The correlation seen for these parameters is caused by the construction of the model. We chose μ_1 and σ_1 to be a priori dependent, as we recall from Figure 3.1. This dependency is reflected in the posterior distribution. The same holds for μ_2 and σ_2. Furthermore, it appears to be some positive correlation between μ_1 and several of the c_k^s parameters, and between μ_2 and several of the τ_k parameters. These correlations are also caused by the structure of the model. Looking at the left panel in Figure 5.4, we see that there generally is some negative correlation between c_i^s and τ_i. Since the c^s and τ are *a priori* independent, the correlation seen in the posterior must be caused by the likelihood function of $\boldsymbol{\theta}$. We recall that the model parameters c^s and τ explain some of the same observed behavior. As c^s increases, the individual will tend to commit to a jar early. But as τ increases, the person will also tend to commit earlier. Therefore, a person that exhibits many early commitments can either have a large c^s and a moderate τ, or he can have a large τ and a moderate c^s. This may cause some negative correlation between c_i^s and τ_i in the posterior density. In Figure 5.5, we show scatter plots of some selected pairs of parameters. We can clearly see the positive correlation between μ_1 and σ_1 and between μ_2 and σ_2. We also see that there is some negative correlation between c^s and τ, especially for individual 100.

Let us look at the parameter estimates we obtain from this sample. We remove the first $2 \cdot 10^5$ iterations as a burn-in period. The histograms of the remaining sample are displayed in Figure 5.6. These histograms show the (empirical) marginal distribution of the parameters. By taking

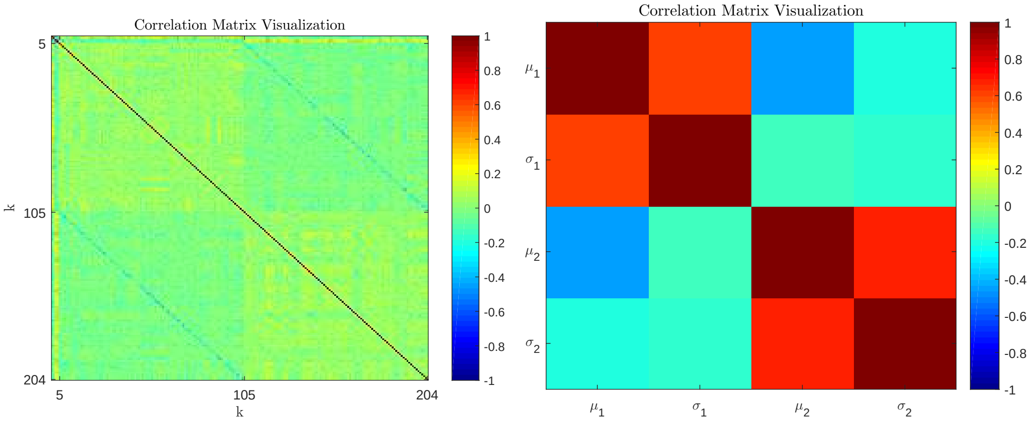

Figure 5.4: Simulation 1: The left panel visualizes the correlation matrix for all the parameters. The right panel has zoomed in on the top left corner of the correlation plot in the left panel. It shows the correlation matrix of the hyperparameters only.

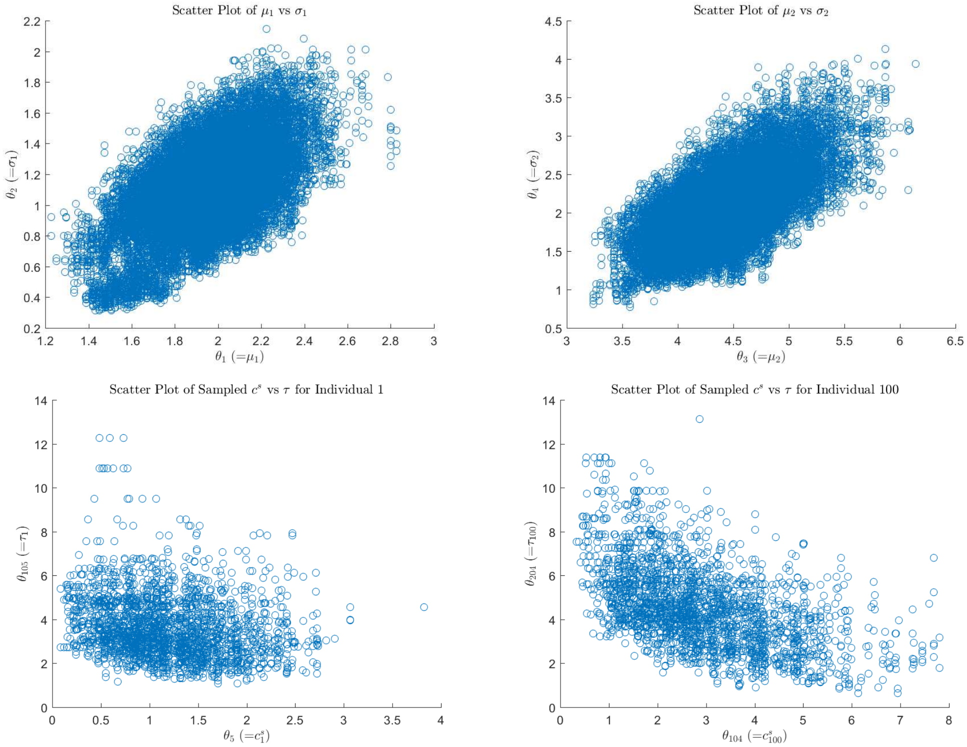

Figure 5.5: SIM1. Scatter plot of pairs of parameters. In the top left panel we have μ_1 versus σ_1 and in the top right we have μ_2 versus σ_2. In the bottom left and bottom right we have c_1^s versus τ_1 and c_{100}^s versus τ_{100}, respectively.

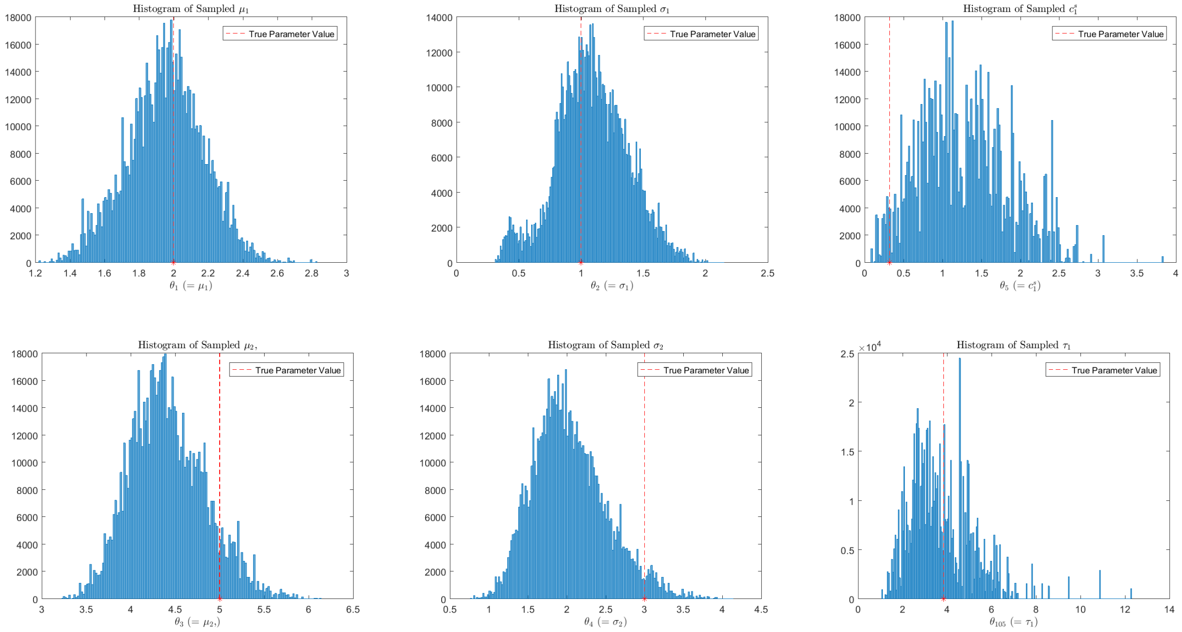

Figure 5.6: SIM1. Histograms of some of the parameters after removing the burn-in period. The true parameter values that were used to generate the individuals and their responses are plotted as red horizontal lines. These histograms represent the shape of estimated marginal distributions of the posterior $p(\boldsymbol{\theta}|y, x, r)$ given in (4.6).

Table 5.2: Summary statistics for some marginal posterior distribution in simulation study 1, 3, and 4. From simulation 1 to simulation 3, we double the number of individuals. From simulation 1 to simultion 3, we double the number of taks per individual.

		Sample Mean (Sample SD))		
θ_k	True	SIM 1	SIM 3	SIM 4
μ_1	2.0000	1.9517 (0.2209)	1.8543 (0.1379)	1.8811 (0.1779)
σ_1	1.0000	1.0904 (0.2850)	0.9678 (0.1727)	1.0995 (0.2118)
μ_2	5.0000	4.4157 (0.4221)	4.9354 (0.3647)	4.9321 (0.3536)
σ_2	3.0000	2.0387 (0.4709)	2.6645 (0.4154)	2.5256 (0.3475)
c_1^s	0.3178	1.2755 (0.5765)	1.1225 (0.5081)	1.1570 (0.5664)
τ_1	3.8371	3.7476 (1.4606)	3.8785 (1.5321)	3.8673 (1.1590)
Sample Size	-	$0.8 \cdot 10^6$	$0.6 \cdot 10^6$	$0.8 \cdot 10^6$

the sample averages, we get the empirical Bayes estimates of the parameters. These estimates and the estimates of the SD of the marginal distributions are shown in Table 5.2. Based on this table and the histograms in Figure 5.6, it seems that by using 100 individuals and four tasks per individual, we are able to estimate the hyperparameters reasonably accurately. In order to visualize how well we are able to retain all the model parameters, we have plotted the true parameters versus the estimates in Figure 5.7. From this plot, we can clearly see the "shrinkage towards the mean" effect of the hierarchical model. The estimates are somewhat squeezed vertically towards the estimated hyperparameter that specifies the mean of the prior distribution. In Figure 5.8, we see a scatter plot of the individuals in the data set; the true c^s is plotted against the true τ for each individual. In the same plot, we have also included the a scatter plot of the individuals based on their estimated parameters. This figure also clearly illustrates the shrinkage effect. The individuals that have extreme parameters are pulled towards the estimated mean of the prior. So individuals with a large c^s tend to get strongly pulled towards $\hat{\mu}_1$, and those with large τ are likely to be pulled towards $\hat{\mu}_2$. We note that the sample size probably is somewhat slim. Even though it contains $8 \cdot 10^5$ values, many of these are duplicates or nearly duplicates that give little extra information. Since the mixing is reasonably bad, it takes a very large sample to represent the distribution well.

Now, we move on to simulation number 2. As we saw in simulation number 1, the acceptance rates were somewhat small for the hyperparameters and somewhat large for the individual-specific parameters. Let us tune the M-H algorithm by decreasing the spread of the former and increasing the spread of the latter. Specifically, we set $\boldsymbol{\epsilon} = (0.5, 0.5, 0.5, 0.5, 2, 2)$ instead of $\boldsymbol{\epsilon} = (1, 1, 1, 1, 1, 1)$. We run the M-H algorithm on the exact same data as in simulation study number 1. That is, we have the same individuals [2] and the same responses for each individual. We want to see how this affects the the acceptance rate, mixing and convergence of the algorithm. We start the chain in the same state as for simulation 1 and we run the same number of iterations. The resulting trace plots are shown in Figure 5.9. This tuning does change the acceptance rates. For μ_1, σ_1, μ_2, and σ_2 they have increased to 0.16, 0.25, 0.12, and 0.26 respectively. The average acceptance rate for c^s and τ have decreased to 0.44 and 0.31, respectively. Com-

[2] An individual in this context means a pair (c^s, τ)

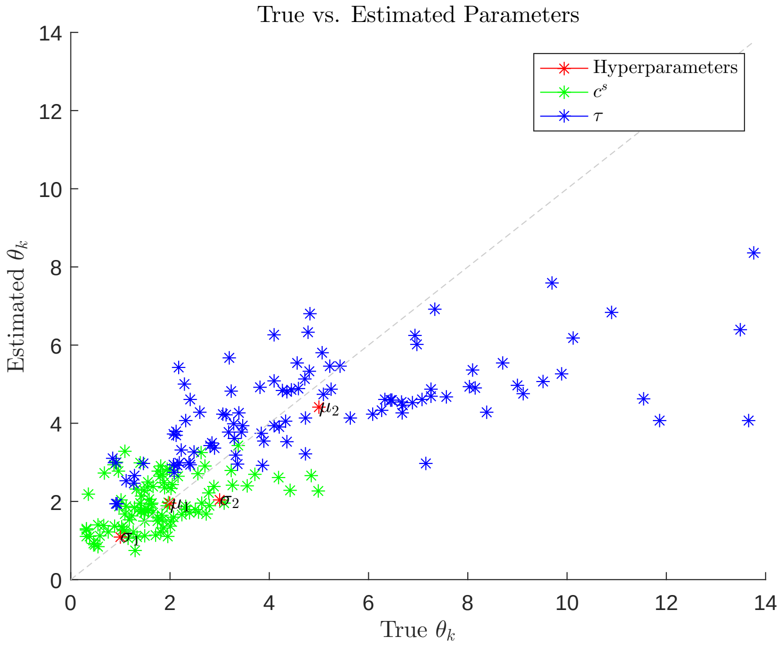

Figure 5.7: SIM1. The true parameters that generated the responses versus the estimates obtained through the Bayesian analysis in simulation study number 1. We can clearly see the "shrinkage" effect on the c^s parameters. All the individuals' sampling costs estimates have a value that is close to the estimated mean of the prior.

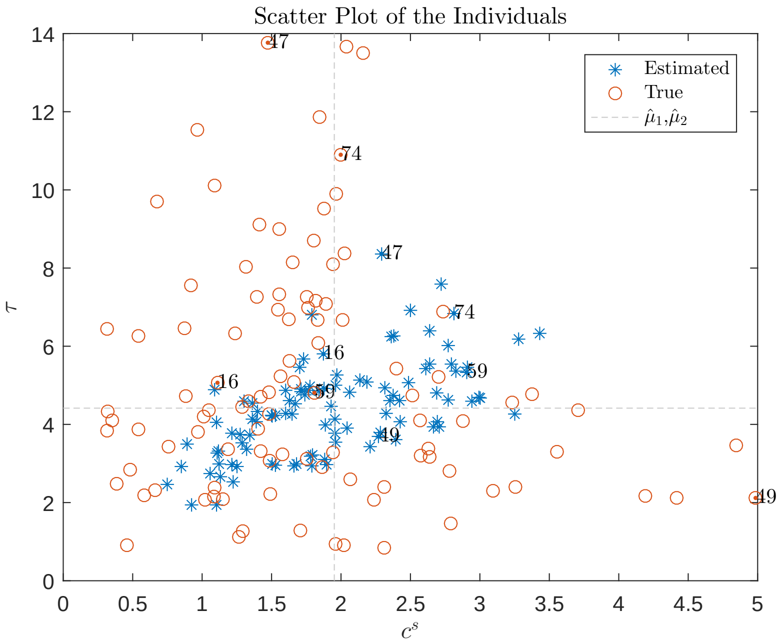

Figure 5.8: SIM1. A scatter plot of the individuals and the estimated individuals in simulation number 1. We have included the ID on some of the individuals in order to show how how the estimated parameters are compared to the true parameters.

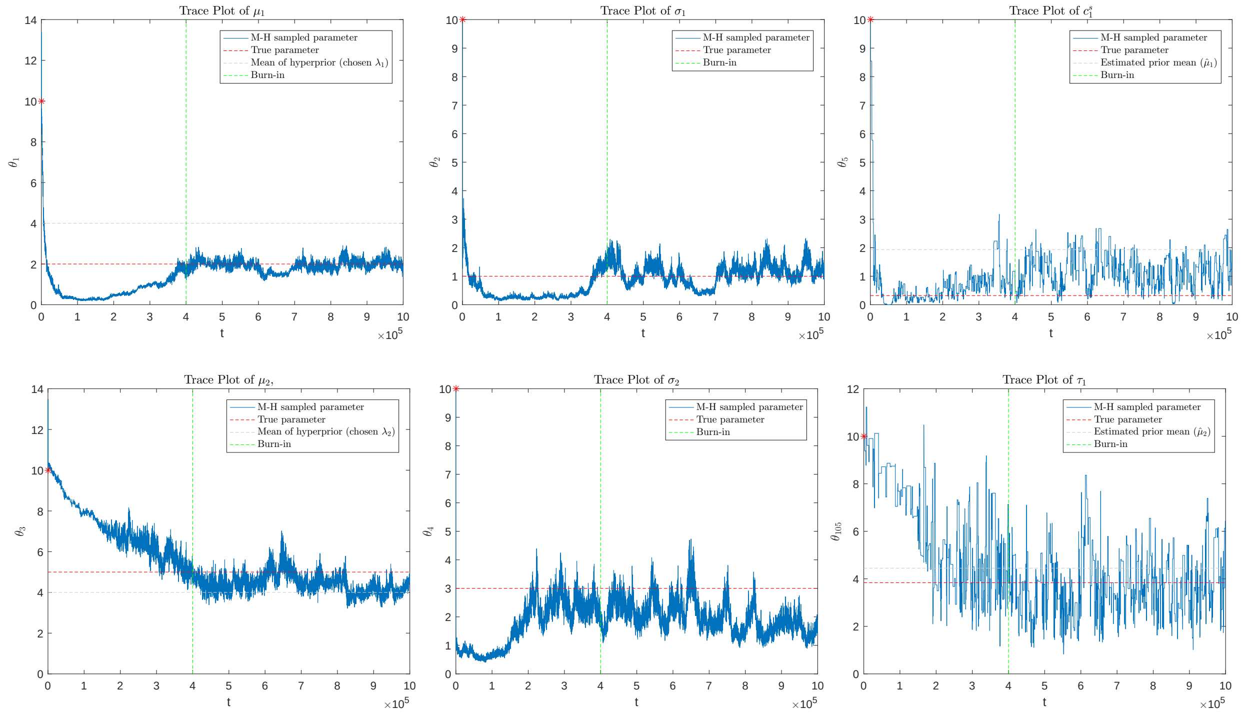

Figure 5.9: SIM2. Trace plots for some of the parameters in simulation study number 2.

paring the trace plots of simulation 1 and 2 in Figure 5.2 and 5.9 respectively, it does not seem like the tuning improved the mixing. On the other hand, it turns out that the "improvement" in the tuning actually slowed down the rate of convergence. It takes about $4 \cdot 10^5$ iterations before stationarity appears to be reached. What we take home from these two simulations, is that the tuning may greatly affect the rate of convergence. However, we should be careful drawing general conclusion from this single example.

In simulation 3, we extend the generated data set from simulation 1. We include the same 100 individuals and their responses as in simulation 1, but we also add another 100 individuals sampled with the same hyperparameters as in simulation 1. We would like to see how this improves the parameters estimation. In advance, we expect an improvement in the hyperparameter estimates. This is because we include more individuals in the data set, and this should give us more information about the distribution from which they are sampled. At the same time, we may expect the rate of convergence to decrease in the M-H algorithm, since the number of parameters are almost doubled. The trace plots for the M-H samples obtained from this data set look reasonable similar to those of simulation 1. It appears that it takes longer for the chain to

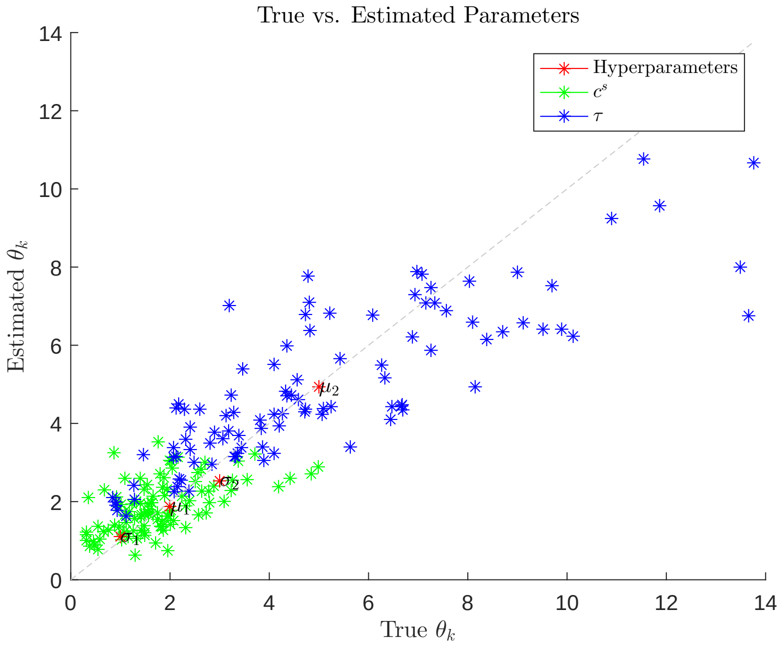

Figure 5.10: SIM4. The true parameters that generated the responses versus the estimates obtained through the Bayesian analysis in simulation study number 4. The shrinkage is less than in simulation number 1.

converge, as anticipated. The estimates of the marginal sample means and SD, after removing an appropriate burn-in period, are printed in Table 5.2. We can see that the estimate of μ_1 actually is somewhat further away from the true value in simulation 3 than in simulation 1. However, we can see that the sample SD of the marginal distribution has decreased, as we expect. For the other hyperparameters we see that the estimates have improved and the SD's have decreased as we move from simulation 1 to simulation 3. This coincides with what we expect.

In simulation 4, we would like to see what happens if we increase the number of tasks per individual. We use the same individuals, tasks and responses as in simulation 1, but we give the individuals four additional tasks and record their responses. In three of these additional tasks, there are two sources of beads, and each with some ratio of white to black beads. In the last additional task, there are four possible sources, where three of the sources have black and white beads, and the fourth has red and black beads. The fourth jar is not the true source in the last task, so no red beads are displayed. The four extra tasks are the same for all the individuals. With respect to the rate of convergence and mixing, the trace plots of simulation 4 look fairly similar to those of simulation 1. We expect an improvement in the individual-specific parameters since more information is gathered from each individual. In turn, we may expect this to improve the hyperparameter estimates, because the more precise information about the individuals should lead to more precise information about the population they come from. The values for some of the parameters estimates are printed in Table 5.2. The estimates of the hyperparameters are reasonably similar as those in simulation 1, and the sample SDs are slightly smaller than in simulation 1. In Figure 5.10, we have plotted the true parameters versus the estimates. We can clearly see that the shrinkage effect is smaller than in simulation 1, especially for the τ parameters. It appears that the individual-specific parameters in general are attained more accurately in simulation 4 than in simulation 1, as expected when more information is gathered from each individual.

In simulation 5, we want to check how sensitive the model is to the hyperprior. We use the same individuals and responses as in simulation 1. However, we make changes to the Bayesian analysis by changing the values of the parameters in the hyperprior. More specifically, we decrease λ_1 from 0.25 to 0.1 and λ_2 from 0.25 to 0.1. This makes the hyperprior less informative, as the variance is increased. It turns out that this modification has very little effect on the parameter estimates. As seen in Figure 5.11, the marginal posterior densities are almost identical in simulation 1 and simulation 5.

This simulation study has given us an impression of how well the H-H algorithm works and how accurately we are able to retain the true model parameters. We have seen that the mixing is reasonably bad. Consequently, we need to perform many iterations of the M-H algorithm. The bad mixing may be due to a poor choice of proposal distribution for the M-H algorithm. We have also seen that we can make good inference about the hyperparameters when the data set consists of 100 individuals with four tasks per individual. The real data set that we analyze in this paper consists of 73 individuals who have performed four tasks each. There is reason to believe that this is enough data to make good inference about the population. However, with only four tasks per participants, the individuals with extreme parameters are not so easily detected. We are now ready to take a look at the data set of real human beings performing the beads task.

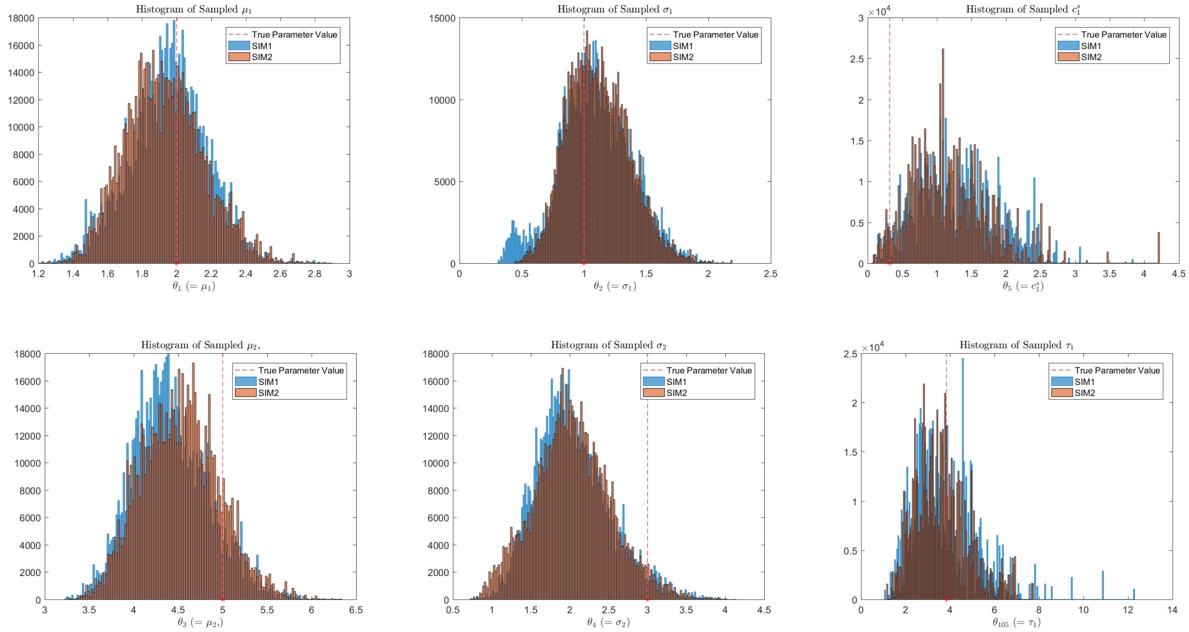

Figure 5.11: Simulation 1 versus simulation 5. We can see that the change in the parameters in the hyperprior have very little effect on the posterior distribution.

Chapter 6

Data Set and Results

In this chapter, we fit the models to experimental data from the Department of Psychology at the "Norwegian University of Science and Technology". The data set consists of the responses of 73 participants performing the tasks in Table 1.1. To our knowledge, none of these participants have been diagnosed with psychosis. We start by looking at the raw data set. Then, we proceed by estimating the parameters under the four models and dicuss what we observe.

6.1 The Raw Data

In this section, we present the raw data from a group of people performing the experiment. We recall that there are two types of responses that are recorded in this experiment. The first type has to do with the DTD part of the experiments, i.e. the number of beads drawn before a commitment to a jar is made and which jar that is chosen. The second type has to do with the DTC part of the experiment, and it consists of the likelihood estimates for each of the jars after each bead is drawn.

Figure 6.1 visualizes the all DTD responses for each of the four tasks. The x-axis shows the i-th bead that is drawn in the task. The left y-axis represents how many participants that responded after seeing the i-th bead, and the vertical bars are associated with this axis. The color of the bars provide information about which of the jars that was chosen. The right y-axis is associated with the solid lines, which are the true probabilities for each of the jars being the source

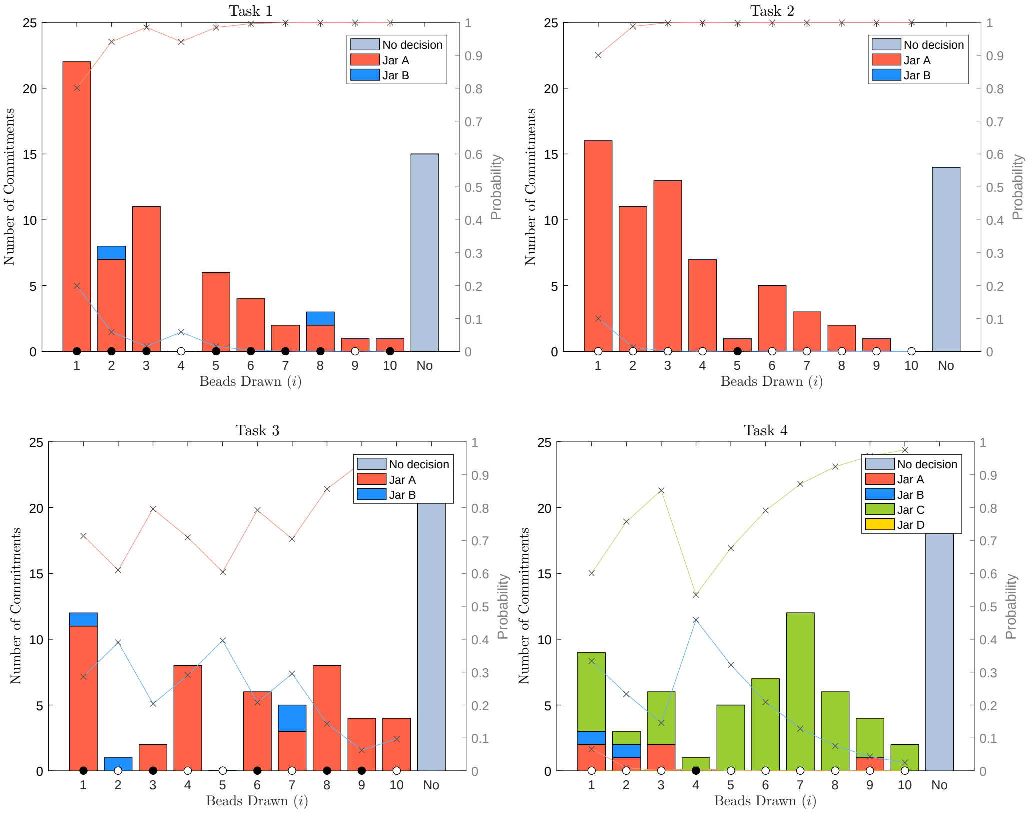

Figure 6.1: DTD responses in the data set. The plot shows how many individuals that responded after the first, second, etc bead was shown. There were in total 73 individuas in the data set. The true probability of each of the jars being the the source is displayed on the right y-axis. Table 1.1 shows the ratios of black, white, and red beads in each of these tasks.

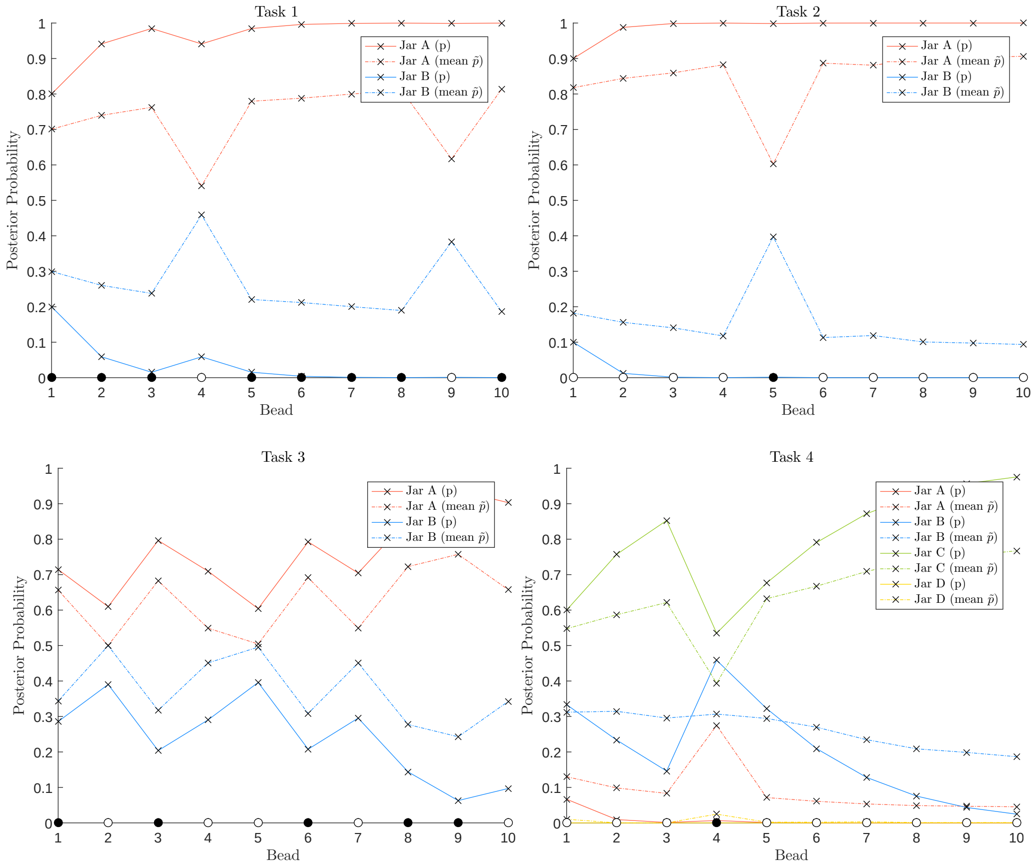

Figure 6.2: DTC responses in the data set. The solid line is the posterior probability of the jar being the source of the beads. The dashed line is the probability estimate, $\tilde{p}$, averaged over all the individuals.[Rewrite this caption!]

when the first i beads have been drawn. The color of the drawn bead is indicated on the x-axis. As we can see, the participants choose the jar with the largest posterior probability of being the in most of the cases where a commitment is made. There are only a few instances where a jar that is not the most likely is chosen. It is also worth noticing that for each of the tasks, there is a large proportion of the individuals that do not commit to any jar during the course of the ten beads. For the first task, we can see that the probability of jar A being the true jar is 0.9 already after the firs bead, and the probability only gets larger as more beads are drawn. Still, as much as 21% of the participants do not commit to any jar in this task.

In Figure 6.2, we present the DTC responses in the data set. The solid line shows the posterior probability of the source being a certain jar. The dashed line, on the other hand, shows the average of the probability estimates provided by the participants. We can see that on aver-

Table 6.1: The empirical Bayes estimates of some of the model parameters. The sample SD's are shown in parenthesis.

	Model			
	HCB	HCBU	HCB-P	HCBU-P
$\hat{\mu}_1$	0.0004 (0.0007)	0.0201 (0.0148)	0.0149 (0.0034)	2.9378 (1.3144)
$\hat{\sigma}_1$	0.0008 (0.0012)	0.0511 (0.0397)	0.0055 (0.0030)	8.6428 (4.2397)
$\hat{\mu}_2$	11.4663 (2.4699)	16.2805 (3.6602)	9.4664 (1.3642)	10.0232 (1.0897)
$\hat{\sigma}_2$	19.3062 (4.6289)	30.9884 (7.4775)	8.9336 (1.8960)	7.3853 (1.3855)
$\hat{c_1^s}$	0.0001 (0.0004)	0.0028 (0.0081)	0.0158 (0.0072)	0.2825 (0.3585)
$\hat{\tau_1}$	2.8622 (1.2025)	3.8465 (2.0799)	6.2683 (1.0397)	9.8286 (2.9514)

age, the probabilities provided by the participants are less extreme than the true probabilities. Furthermore, the participants tend to react strongly when a bead of a different color than its preceding beads shows up. This is for example seen when the fourth bead is drawn in Task 1.

6.2 Model Results and Discussion

We fit model parameters under the HCB, HCBU, HCB-P, and HCBU-P models. To account for the fact that the participants are not required to make a commitment to any jar during the task, we have used that $n_{\text{assumed}} = 20$. We use the M-H algorithm with the tuning parameters as in simulation 1 in Table 5.1. The resulting (empirical) mean and SD of the marginal distribution of some selected parameters are shown in Table 6.1.

Let us start by analyzing the posterior distribution of the model parameters attained under the HCB model. This is the same model as the one introduced by Moutoussis *et al.*. In Figure 6.3, we present the trace plots for the hyperparameters μ_1, σ_1, μ_2, and σ_2, and the trace plots for c^s and τ for one of the individuals. After checking the trace plot of every single parameter, it looks like convergence is attained after about $4 \cdot 10^5$ iterations. The corresponding histograms after removing the burn-in period are shown in Figure 6.4. In agreement with Moutoussis *et al.*, we find that the mean of the sampling costs (μ_1) is near zero. Furthermore, we find that the mean of the decision noise parameters (μ_2) is of the same magnitude as the one obtained by Moutoussis *et al.*. Specifically, we find $\hat{\mu}_2 \approx 11.47$, while Moutoussis *et al.* obtained the value

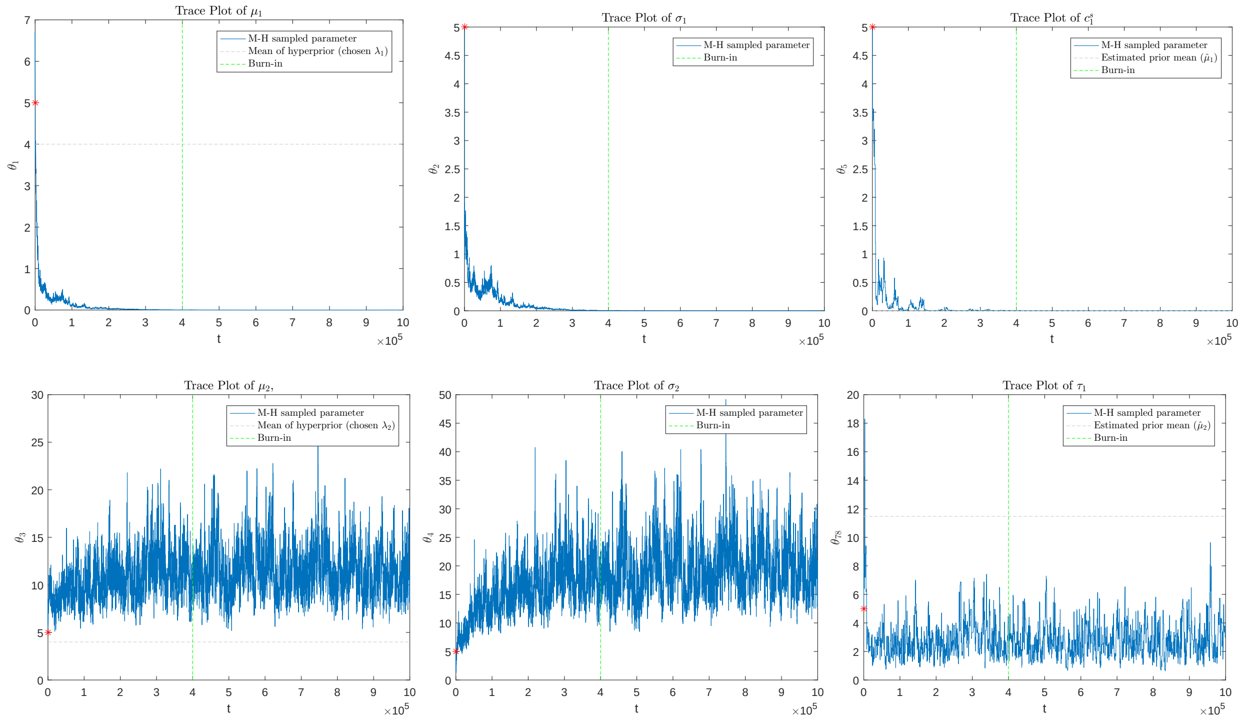

Figure 6.3: Trace plot for HCB model results.

5.09 for the control group and the value 12.07 for the paranoid group.

Even though each individual only responds to four tasks, it is interesting to take a look at the estimates of the individual-specific parameters. In Figure 6.5, we show the histograms of the $\hat{c}^s$ and $\hat{\tau}$. All the individuals get an estimated sampling cost that is less than $1.3 \cdot 10^{-3}$, which is basically zero since it has negligible effect on the action value for sampling . For the decision noise parameter τ, the estimates are ranging from 0.0011 to 60. We can see that there is a large group of individuals that obtain a τ which is very close to zero. 11 individuals have $\hat{\tau}$ with a magnitude around 0.0011. A person with $\tau = 0.0011$ is basically behaving as an IBA. Taking a closer look at the 11 individuals with smallest τ parameters, we find that all these individuals sampled 10 all beads without committing to any jar in the end, in each of the four tasks. It makes sense that these get very small estimates of τ, because an IBA with $c^s = 0$ would with probability 1 sample through all the beads without making any commitment to a jar. The prior, however, is preventing these individuals from obtaining $\hat{\tau} = 0$.

The individuals have estimated sampling costs that are virtually zero. The decision noise pa-

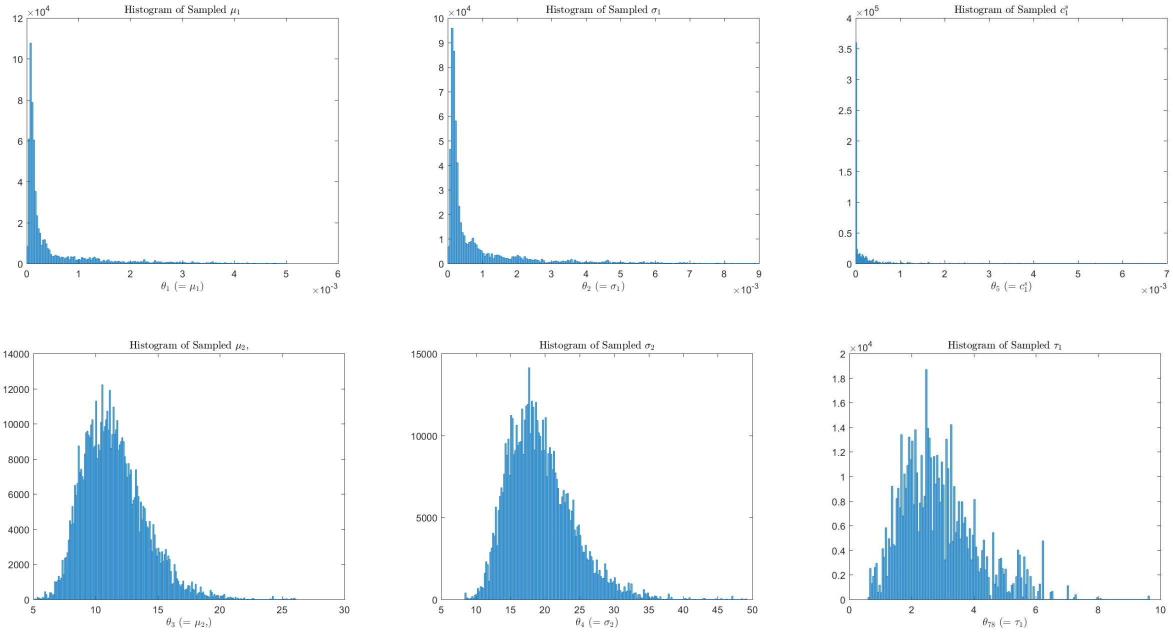

Figure 6.4: HCB model restult. Histograms of some of the parameters after removing the burn-in period.

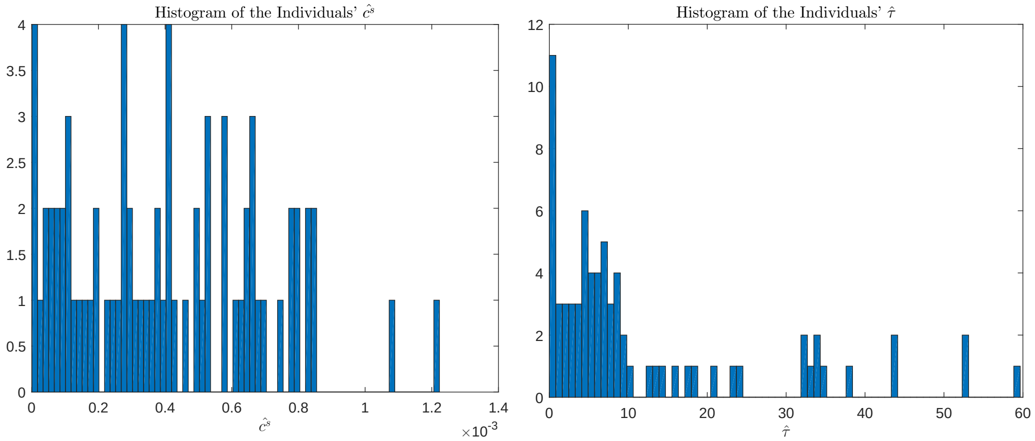

Figure 6.5: HCB model results. From left to right, the plots respectively show histogram of the estimated sampling costs and histogram of estimated decision noise parameters.

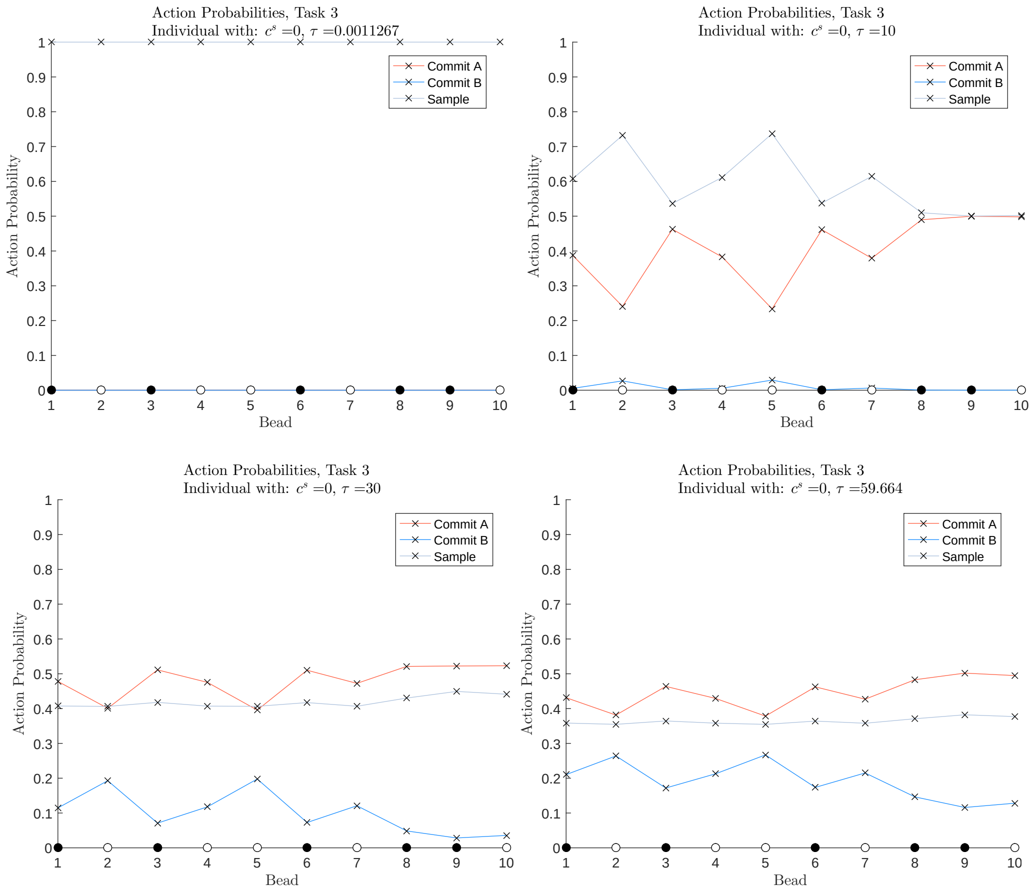

Figure 6.6: Action probabilities for different individuals under the HCB model. The task is Task 3 in Table 1.1. These plots help us interpret the values of $\hat{\tau}$ that we obtain in the data set.

rameters are ranging from 0.0011 to 60. So what does this imply on the individuals' behavior? We are curious to know if $\tau = 60$ is so large that the agent gives a weight of $1/3$ to each action and therefore makes arbitrary choices. We recall that an agent chooses stochastically between each action, and the probability of each action is weighted based on the action values. Let us take a closer look at the expected behavior of agents with $c^s = 0$ for different values of τ. In Figure 6.6, we have plotted the action probabilities in Task 3 for individuals with $\tau = 0.0011$, $\tau = 10$, $\tau = 30$, and $\tau = 60$. As we can see, the agent with $\tau = 0.0011$ deterministically chooses to sample another bead in each state. As τ increases, the agent becomes less prone to sampling. Instead, the probability of choosing a jar increases. The probability of choosing the more likely jar is always larger than the probability of choosing the less likely jar, as we known theoretically from 3.10. For an individual with $\tau = 60$, the probability of sampling after the first bead is shown (in Task 3) is about 0.35, which means that the probability of committing is about 0.65. This probability

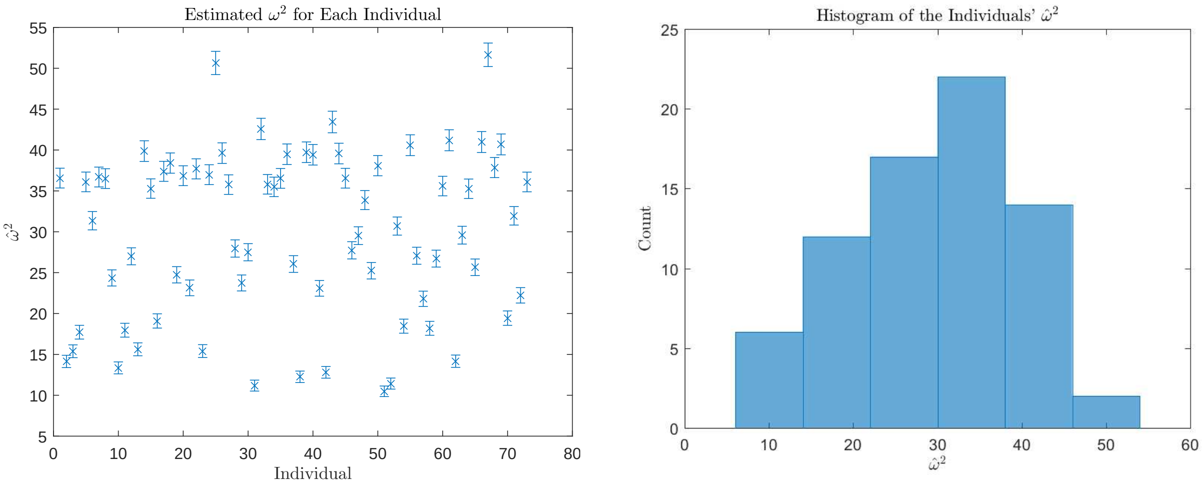

Figure 6.7: In the left panel, we see the $\hat{\omega}^2$ as well as their 95% confidence intervals. In the right panel, we have plotted a histogram of the esimated $\hat{\omega}^2$.

remains reasonable stable throughout the task. However, the weight given to each of the jars fluctuates as new beads are drawn. After the first bead is drawn, the probability of choosing the jar that is more likely of being the source is about 0.45. These plots give us some understanding of the $\hat{\tau}$ that we obtain in the data set.

Next, we look at the MLE estimate we obtain for $\boldsymbol{\theta}^* = (\omega_1, ..., \omega_{N_{\text{ind}}})$ in the HCB-P and HCBU-P models. In the left panel in Figure 6.7, we have plotted each $\hat{\omega}_i^2$ and their estimated 95% confidence intervals based on the observed Fisher information. All $\hat{\omega}$ lie within the range $(3.24, 7.19)$. As we recall from (3.12), we have assumed that the noise term is additive and normally distributed. Therefore, if the assumption is reasonable, we should see that the residuals are normally distributed with a mean equal to zero and an variance equal to ω_i for each individual i. In Figure 6.8, we show residual plots for two arbitrary individuals. The residual plots for the rest of the individuals are reasonably similar to these. We can see that the normality assumption is not very appropriate. The grey vertical lines show where a new trial starts. For the two first trials, we can see that the residuals increase in absolute value as more beads are drawn. This trend is caused by our choice of using the log-odds function for mapping the ρ to the real line; when ρ departs from 0.5 towards 0 or 1, the difference between $\log(\frac{\rho}{1-\rho})$ and $\log(\frac{\rho-\Delta}{1-(\rho-\Delta)})$ increases. As a result, an individual that for example underestimates the probability ρ with $\Delta = 0.05$, the effect of this error is much larger (on the transformed scale) if the $\rho = 0.90$ than if $\rho = 0.5$. Therefore,

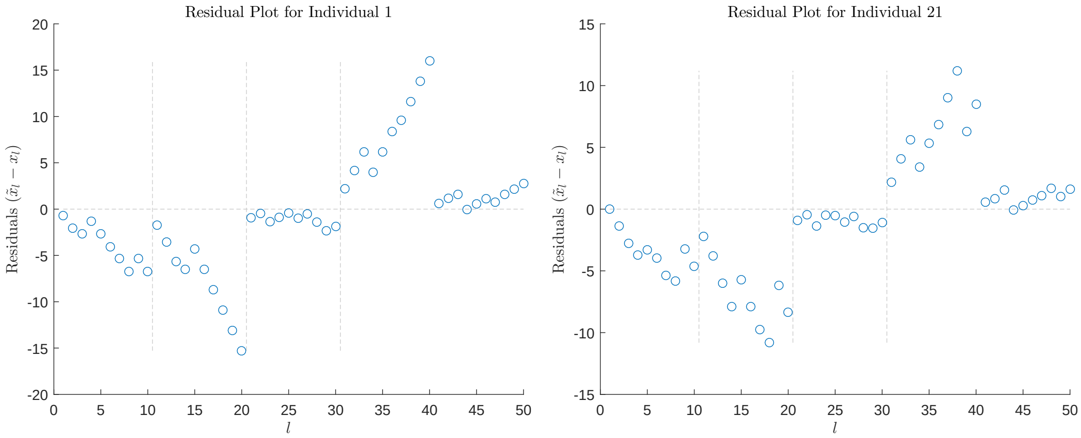

Figure 6.8: Residual plots.

the error in probability estimate that is made when the true probability is close to 0 and 1 has a very large impact on $\hat{\omega}$.

Let us compare the results obtained in the four models. Comparing the hyperparameter estimates in Table 6.1, we note the following. First of all, we see that when moving from the HCB model to the HCB-P model, $\hat{\mu}_2$ decreases. The same is seen when moving from the HCBU to the HBCU-P model. In other words, the expected value of the noise parameter decreases. This may be because the decisions made by the participants are more reasonable when we consider how likely they *think* that each jar is. Secondly, we note that σ_2 hat is reasonably large in all the models, especially for the HCB and HCBU models. This suggest that there is a large variety within the population. Thirdly, is appears that the sampling costs actually play a role in the HCBUP model, in contrast to the other three models.

Chapter 7

Conclusion and Future Work

In this report, we have introduced four alternative models that attempt to explain the decision-making process for individuals performing the beads task. These models use the IBA as a reference and adds noise to his behavior. In two of the models, the HCB and the HCBU models, there is one parameter (c^s) that captures the eagerness to make a commitment to a jar and one parameter (τ) that accounts for the randomness in choice for each individual. The two other models, the HCBUP and the HCBP models, additionally tries to take into consideration that the individuals have noise when they calculate the posterior probability of each of the jars being the source of the displayed beads. This is done by imposing yet another individual-specific parameter in the model (ω).

In the Background section, we discussed several reasoning biases that may explain why delusions are formed and maintained. Now, let us take a look at how the models we have introduced in this report may account for - or fail to account for - two of these reasoning biases. We recall that the JTC bias is the tendency to make hasty decisions. The models we have introduced may explain why some people make hasty decisions by the fact that they have a large c^s. However, a large decision noise parameter may also boost hasty decisions, since the probability of choosing to sample another bead decreases as τ increases. When a large decision noise parameter is the cause of hasty decisions, the models predict that there should be more wrong commitments. Here, when we say "wrong commitment", we refer to choosing a jar that is not the more likely for being the source of the drawn beads. We suggest modifying these models such that large decision noise has a less impact on hastiness. Perhaps there should be separate procedures for determining 1) whether or not to decide on a jar or keep sampling and 2) which of the jars to pick. The LA account, in contrast to the JTC account, says that the hastiness relative to a control group may vanish under ambiguity. This is because multiple alternatives may seem plausible enough, causing none of them to be chosen and instead more evidence to be sought. Our models, however, do not justify the LA bias by design. If a person has a greater tendency of making a fast decision than another person in the two-jar case (either as a result of larger c^s or τ), the models predict that he will have a greater tendency of making a faster decision than the other person in the four-jar case as well.

When applying our model of the probability estimates, $p(\tilde{\rho}|\omega)$, to the real data set, we discovered that it had some flaws. First of all, we discovered that it had some undesirable effects when the true probability for either of the jars was close to 0 or 1. Secondly, it appeared that we should include a parameter that takes bias into account; it seemed like the participants were conservative in their probability ratings. Therefore, we suggest making a new model which includes a bias parameter that systematically pulls the probability estimate towards 0.5.

In our report, we did not compare the fit of the four models. We suggest performing model selection to figure out which of the four models that is better at accounting for the decision-making process of human beings. Lastly, we suggest fitting the models to a healthy control group and a group of deluded patients in order to see if there are any significant group differences.

Appendix A

Results

In this appendix, we present some plots of the results ob applying the HCBU, HCBP, and HCBUP models to the real data set. These plots are equivalent to those presented for the HCB model in the main text.

A.1 HCBU Model

A.2 HCBP Model

A.3 HCBUP Model

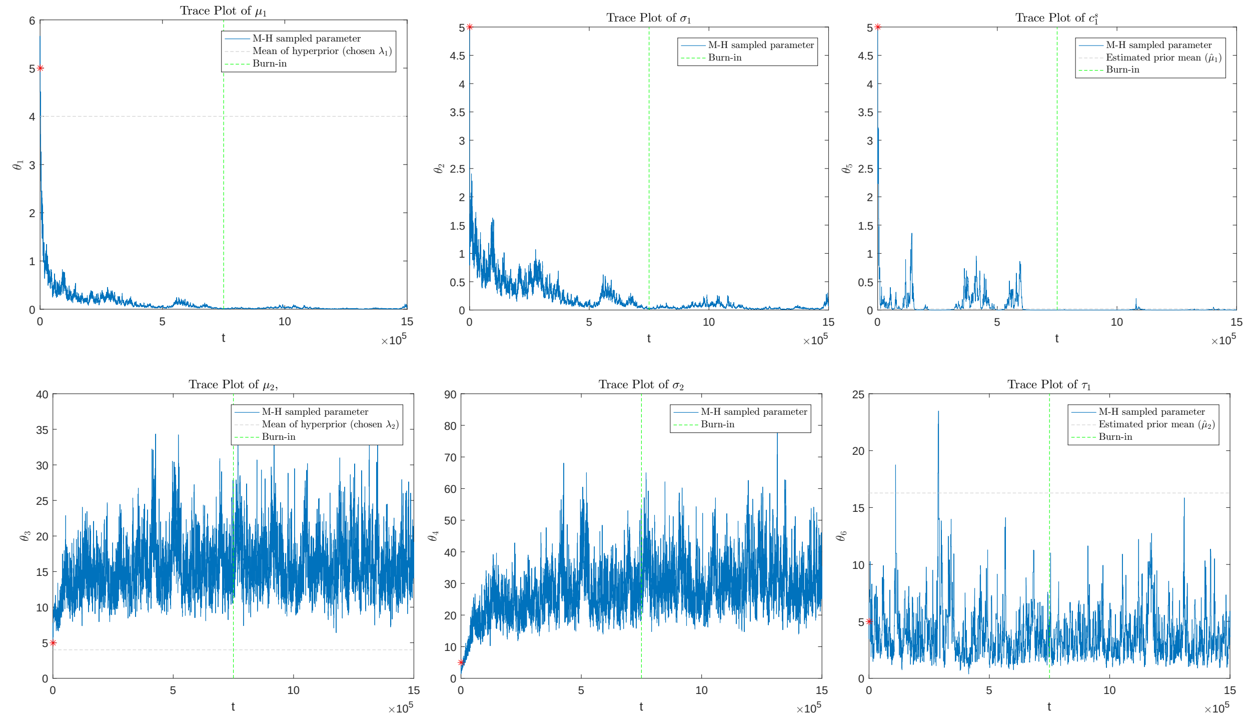

Figure A.1: HCBU model reuslts.

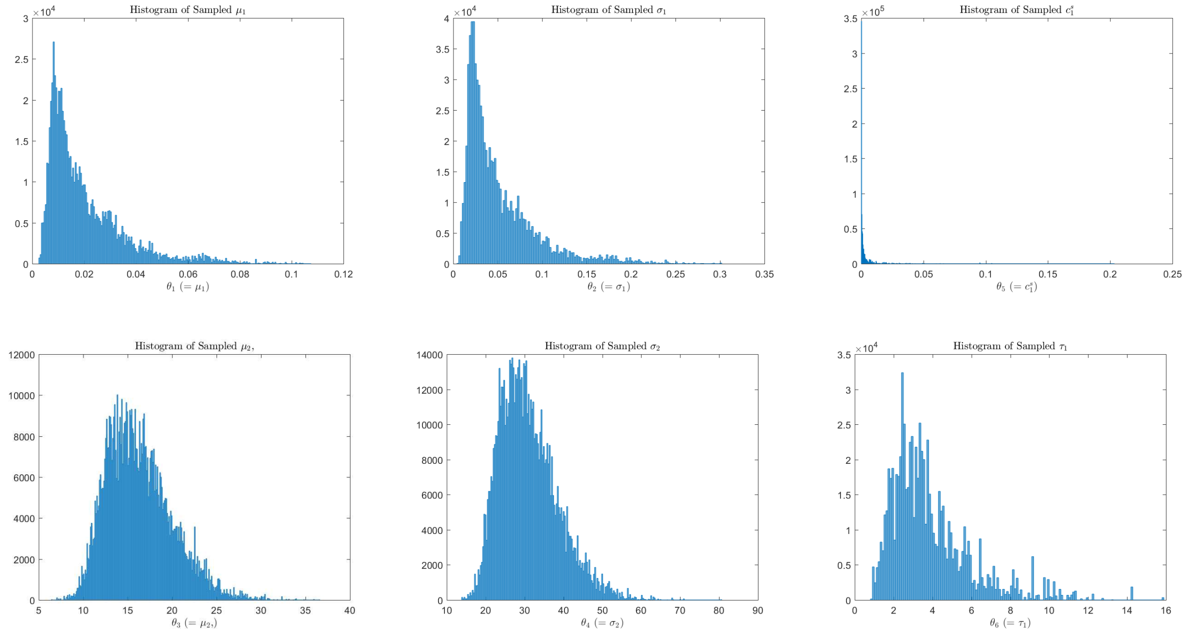

Figure A.2: HCBU model restult. Histograms of some of the parameters after removing the burn-in period.

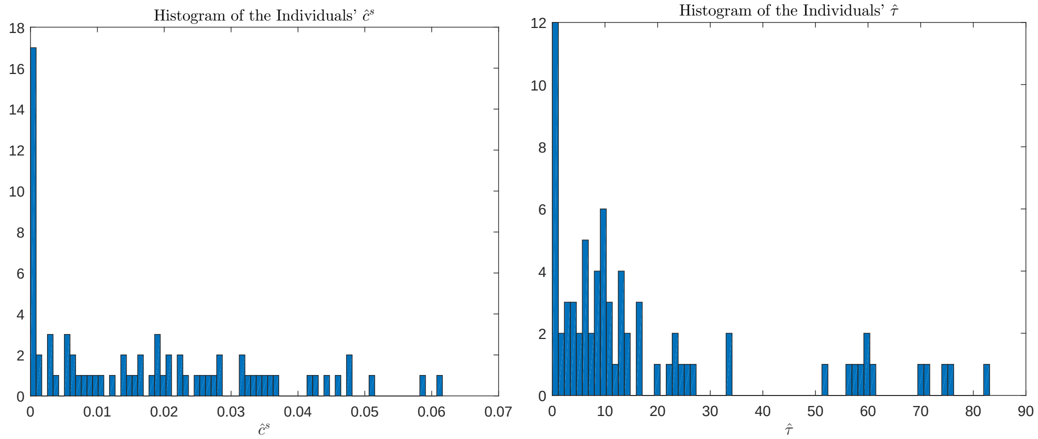

Figure A.3: HCBP model results.

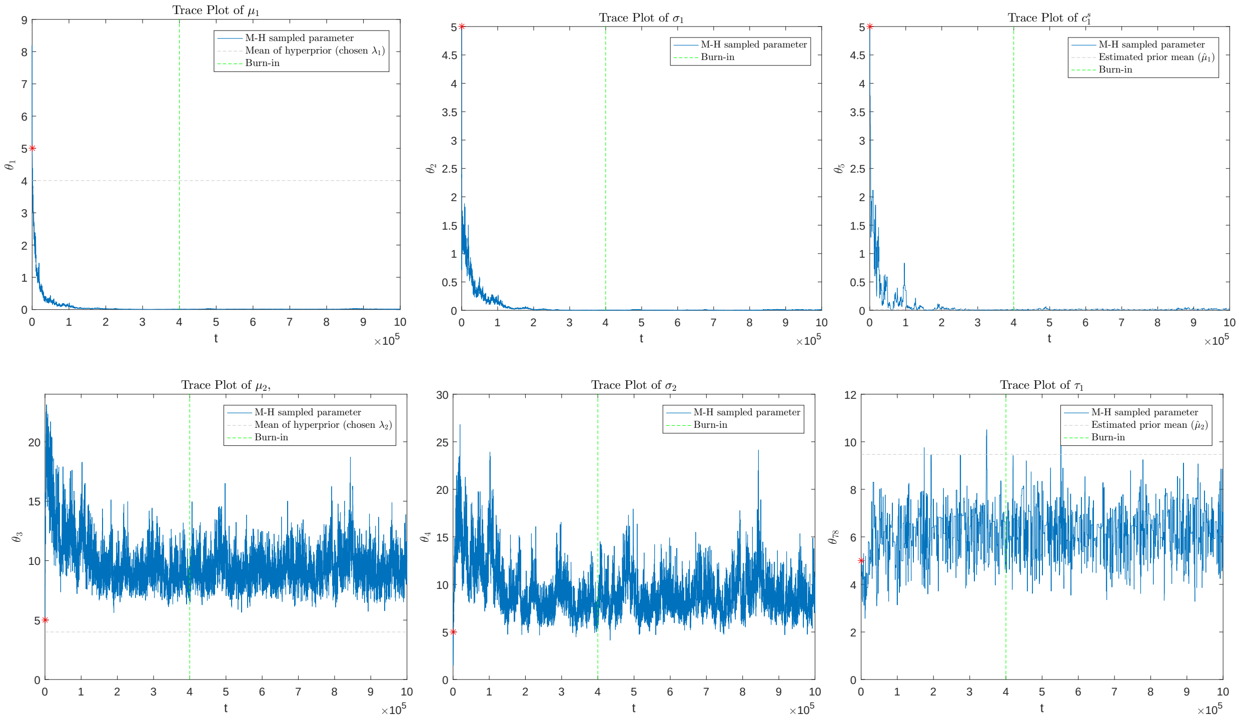

Figure A.4: HCBP model reuslts.

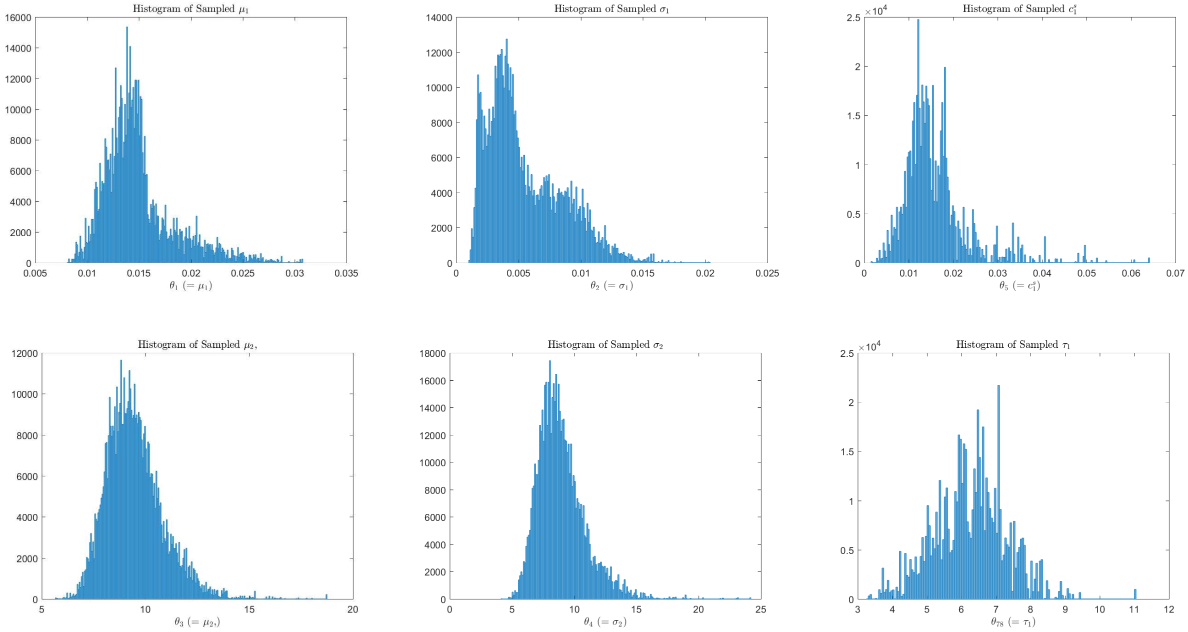

Figure A.5: HCBP model restult. Histograms of some of the parameters after removing the burn-in period.

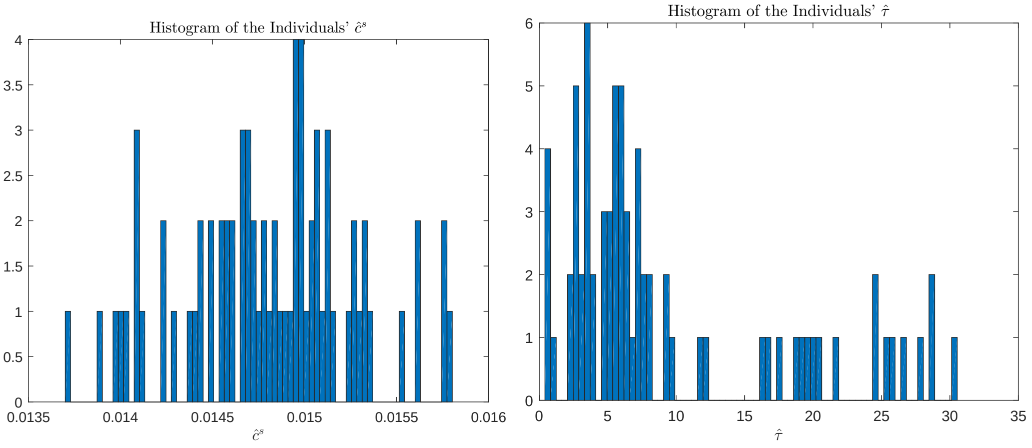

Figure A.6: HCBP model results.

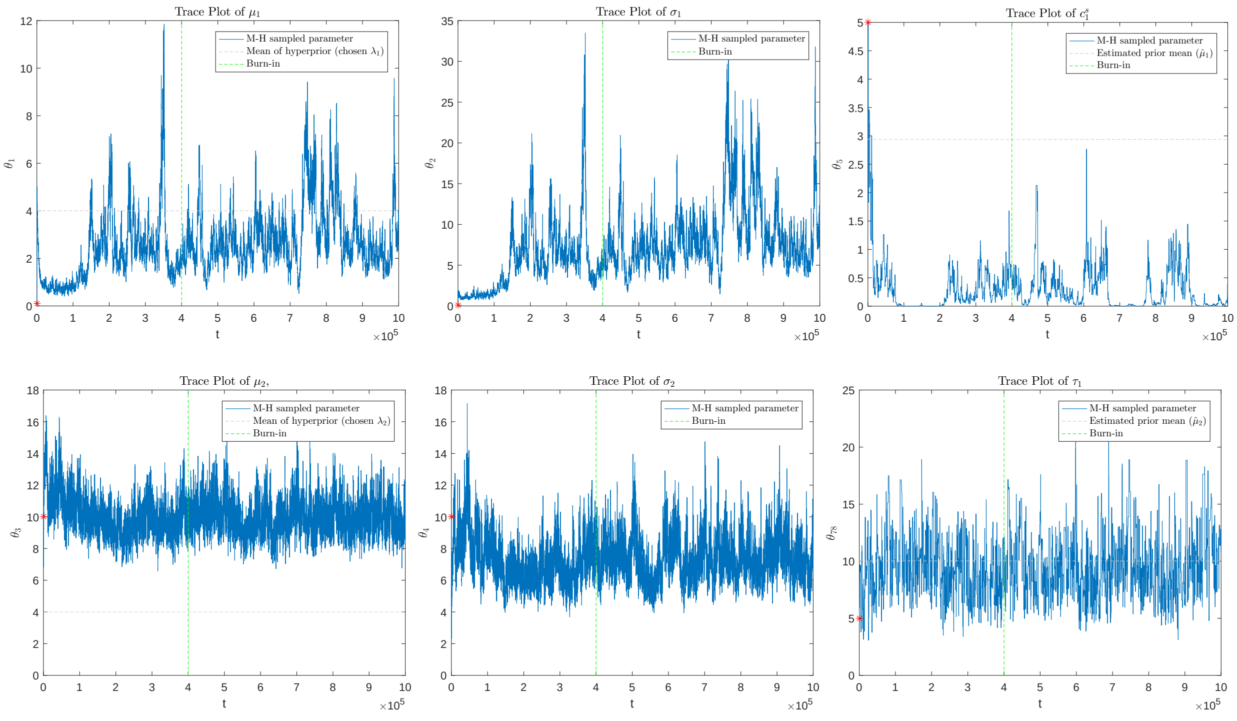

Figure A.7: HCBP model reuslts.

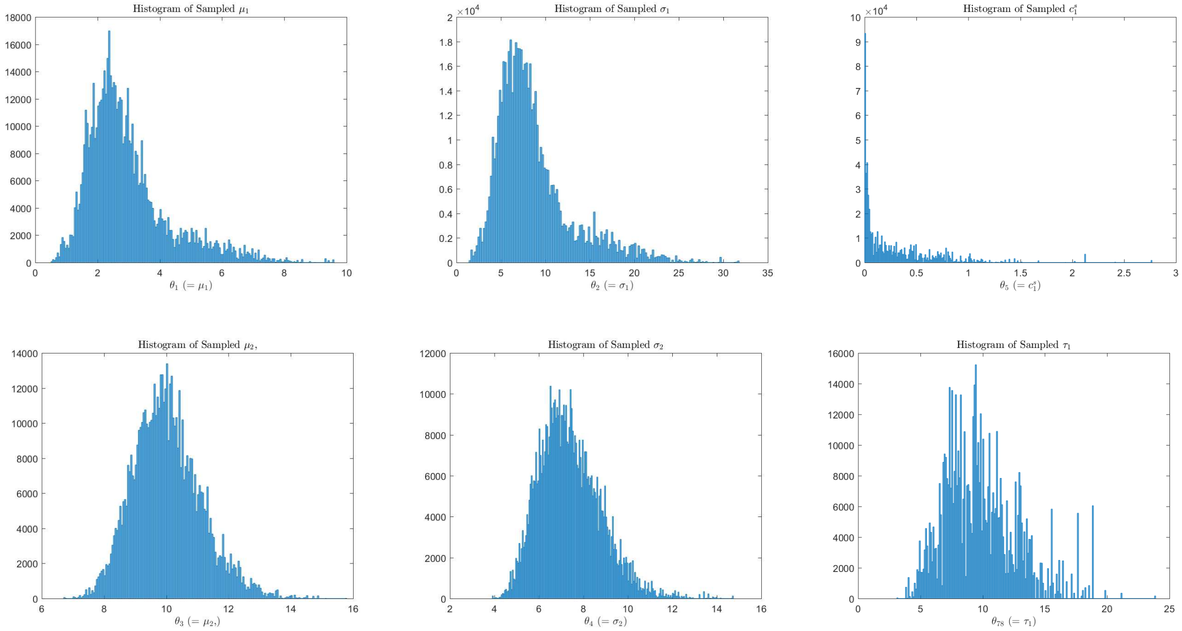

Figure A.8: HCBU-P model restult. Histograms of some of the parameters after removing the burn-in period.

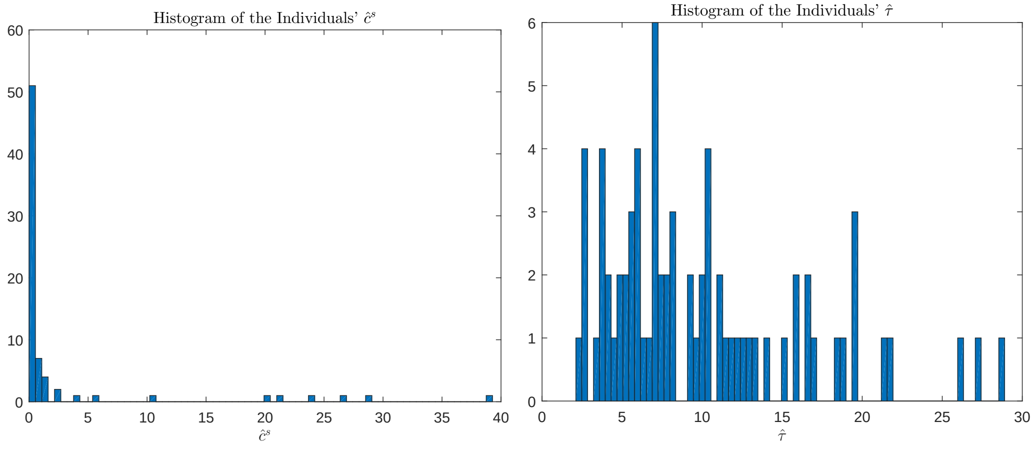

Figure A.9: CAPTION. HCBUP model results.

Appendix B

Acronyms

BADE Bias Against Disconfirmatory Evidence

BR Bayes' Rule

DTD Draws to Decision

IBA Ideal Bayesian Agent

JTC Jumping to Conclusion

LA Liberal Acceptance

MCMC Markov chain Monte Carlo

M-H Metropolis-Hastings

MLE Maximum Likelihood Estimation

SZ Schizophrenia

HCB Hierarchial Costed Bayesian

HCBU Hierarchial Costed Bayesian, Unaware

HCBP Hierarchial Costed Bayesian with Probability estimates

HCBUP Hierarchial Costed Bayesian, Unaware with Probability estimates

Bibliography

[1] American Psychiatric Association et al. *Diagnostic and statistical manual of mental disorders (DSM-5®).* American Psychiatric Pub, 2013.

[2] Siddhartha Chib and Edward Greenberg. Understanding the metropolis-hastings algorithm. *The american statistician*, 49(4):327–335, 1995.

[3] Peter Congdon. *Bayesian statistical modelling*, volume 704. John Wiley & Sons, 2007.

[4] Mary Kathryn Cowles and Bradley P Carlin. Markov chain monte carlo convergence diagnostics: a comparative review. *Journal of the American Statistical Association*, 91(434):883–904, 1996.

[5] Robert Dudley, Peter Taylor, Sophie Wickham, and Paul Hutton. Psychosis, delusions and the "jumping to conclusions" reasoning bias: a systematic review and meta-analysis. *Schizophrenia bulletin*, page sbv150, 2015.

[6] Julian J Faraway. *Extending the linear model with R: generalized linear, mixed effects and nonparametric regression models.* CRC press, 2005.

[7] Niall Galbraith. *Aberrant Beliefs and Reasoning.* Psychology Press, 2014.

[8] D. Gamerman and H.F. Lopes. *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference, Second Edition.* Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 2006.

[9] Philippa Garety, Daniel Freeman, Suzanne Jolley, Kerry Ross, Helen Waller, and Graham Dunn. Jumping to conclusions: the psychology of delusional reasoning. *Advances in psychiatric treatment*, 17(5):332–339, 2011.

[10] Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian data analysis*, volume 2. Taylor & Francis, 2014.

[11] G.H. Givens and J.A. Hoeting. *Computational Statistics*. Wiley Series in Computational Statistics. Wiley, 2012.

[12] Steffen Moritz, Todd S Woodward, and Martin Lambert. Under what circumstances do patients with schizophrenia jump to conclusions? a liberal acceptance account. *British Journal of Clinical Psychology*, 46(2):127–137, 2007.

[13] Michael Moutoussis, Richard P Bentall, Wael El-Deredy, and Peter Dayan. Bayesian modelling of jumping-to-conclusions bias in delusional patients. *Cognitive Neuropsychiatry*, 16(5):422–447, 2011.

[14] V.K. Rohatgi and A.K.M.E. Saleh. *An Introduction to Probability and Statistics*. Wiley Series in Probability and Statistics. Wiley, 2011.

[15] William J Speechley and Jennifer C Whitman. The contribution of hypersalience to the" jumping to conclusions" bias associated with delusions in schizophrenia. *Journal of psychiatry & neuroscience: JPN*, 35(1):7, 2010.

[16] Todd S Woodward, Lisa Buchy, Steffen Moritz, and Mario Liotti. A bias against disconfirmatory evidence is associated with delusion proneness in a nonclinical sample. *Schizophrenia bulletin*, 33(4):1023–1028, 2007.