Erasmus University Rotterdam
Erasmus School of Economics
Master Thesis Business Analytics & Quantitative Marketing

Behavioural Artificial Intelligence Technology: Designing a Bayesian Approach and Investigating the Feedback Loop

Maarten Wijnands
Student number: 572086

Supervisor: Prof.dr. Dennis Fok
Second assessor: Dr. Kathrin Gruber
Company supervisor: Prof.dr.ir. Caspar Chorus
Date final version: 23rd February 2022

Abstract

This paper offers an elaborate study on the novel Decision Support System called Behavioural Artificial Intelligence Technology (BAIT). Specifically, the paper investigates the suitability of a Bayesian methodology for BAIT and the effects of BAITs feedback loop. For the first topic, several Bayesian approaches are proposed as alternatives to the current frequentist estimation routines. Our study shows that Bayesian inference offers attractive model assumptions and an elegant methodology to sequentially update the parameters of BAIT. In particular, applying sequential Variational Logistic Regression on real-life decisions with a prior originating from Maximum Likelihood Estimation on a data set of hypothetical decisions is found to be very suitable for the objectives of BAIT. The second part of the paper studies the effect of BAITs feedback loop. Our self-designed simulation framework provides evidence that continuously iterating through the feedback loop is likely to result in an amplification of BAITs recommendations and homogenization of expert behaviour over time. Dependent on the goal of applying BAIT, this may or may not be a desirable outcome. Additionally, this paper explores how these effects are impacted by altering feedback loop conditions, such as the implementation stage of BAIT, expert dependency, expert heterogeneity and group size.

COUNCYL
BEHAVIORAL AI TECHNOLOGIES

Erasmus
University
Rotterdam

Preface

In May 2021, I received an email from Dennis Fok about a vacancy at a TU Delft spin-off called Councyl, which offered a challenging thesis internship in an ambitious start-up environment. The email instantly caught my eye and only a couple of days later, I met with Councyls founders Caspar Chorus and Nicolaas Heyning. Both parties were immediately so enthusiastic that the agreement was settled within a couple of hours. Since that moment Caspar Chorus, Nicolaas Heyning and Dennis Fok have been of the greatest support in my thesis trajectory and my enthusiasm has risen ever since.

First of all, I would like to thank Councyl for offering the opportunity to write my thesis with the company and having continuous faith in my capabilities. Not for a moment, I have felt unappreciated and, even though my project was quite far from the day-to-day business, I have always felt like an important part of the team. A huge part of this can be attributed to Nicolaas Heyning, Annebel ten Broeke, Stella Mulia and Monica Ferraioli. The countless conversations I have had with you have not only provided me with a lot of insights related to my thesis, but also greatly contributed to my happiness at Councyl. In particular, I would like to express my gratitude to Caspar Chorus, since you have always made time for me and offered great guidance throughout the process. I especially admire you for always maintaining a positive mindset and inspiring all the people around you.

Furthermore, I want to express my gratitude to my thesis supervisor Dennis Fok. Because of your excellent lectures in Advanced Marketing Models, it was my personal preference that you would supervise my thesis. I can fairly say that you have lived up to all expectations. Both the meetings and feedback on my thesis have always been extensive and motivating. Your ideas and insights have greatly contributed to the contents of this thesis.

On a personal note, I would like to thank all my friends and family for their unconditional support. Especially, I want to mention my parents, who have always been my beacon of support in all thinkable ways. Words will never be able to describe my immense gratitude towards you.

I am proud of the thesis that I am handing over. It would not have been the same without every single one of you.

Maarten Wijnands
February 16th 2022

Contents

1 Introduction **1**
- 1.1 Decision Support Systems . . . 1
- 1.2 Research goals . . . 2
  - 1.2.1 A Bayesian approach to BAIT . . . 2
  - 1.2.2 The feedback loop in BAIT . . . 3
- 1.3 Research structure . . . 3

2 Behavioural Artificial Intelligence Technology **4**
- 2.1 Background information . . . 4
- 2.2 Case study . . . 5
- 2.3 Problem statement . . . 5

3 Data description **7**
- 3.1 Introduction . . . 7
- 3.2 Special welfare applications . . . 7
  - 3.2.1 Data description . . . 7
  - 3.2.2 Variable description . . . 8
- 3.3 COVID-19 ICU uptakes . . . 9
  - 3.3.1 Data description . . . 9
  - 3.3.2 Variable description . . . 10
- 3.4 Data modifications . . . 12

4 A Bayesian approach to BAIT **13**
- 4.1 Introduction . . . 13
- 4.2 Literature review: Background and motivation . . . 14
  - 4.2.1 Introduction to Bayesian inference . . . 14
  - 4.2.2 Bayesian inference in DSS . . . 14
  - 4.2.3 Bayesian inference in BAIT . . . 15
- 4.3 Literature review: Bayesian methodology . . . 16
  - 4.3.1 Binary behavioural models . . . 16
  - 4.3.2 Prior distributions . . . 16
  - 4.3.3 Batch and sequential estimation . . . 17
  - 4.3.4 Markov Chain Monte Carlo methods . . . 18
  - 4.3.5 MCMC methods for binary models . . . 19

4.3.6 Variational Logistic Regression . . . 19
4.4 Methodology . . . 20
4.4.1 Outlining the various approaches . . . 20
4.4.2 Binary models . . . 21
4.4.3 MLE and Bayesian approaches . . . 22
4.4.4 Formalizing the prior distribution . . . 23
4.4.5 Gibbs sampler for the probit model . . . 26
4.4.6 Metropolis-Hastings sampler for the logit model . . . 27
4.4.7 Variational Logistic Regression . . . 29
4.4.8 Validation and model performance . . . 31
4.5 Results for the special welfare applications . . . 33
4.5.1 Prior parameters . . . 34
4.5.2 Evaluation of prior inputs . . . 35
4.5.3 Bayesian modelling conditions . . . 36
4.5.4 Posterior distributions . . . 36
4.5.5 Model performance . . . 38
4.6 Results for COVID-19 ICU uptakes . . . 41
4.6.1 Prior parameters . . . 41
4.6.2 Posterior distributions . . . 41
4.6.3 Model performance . . . 43
4.7 Conclusion . . . 44

5 The feedback loop in dynamic BAIT **46**
5.1 Introduction . . . 46
5.2 Literature review . . . 46
5.3 Conceptualizing the feedback loop . . . 48
5.3.1 One expert, without BAIT . . . 48
5.3.2 Multiple experts, without BAIT . . . 48
5.3.3 One expert, with BAIT . . . 49
5.3.4 Multiple experts, with BAIT . . . 50
5.4 Methodology . . . 50
5.4.1 Simulation assumptions . . . 51
5.4.2 The investigated effects . . . 53
5.4.3 The investigated conditions . . . 54
5.5 Results . . . 56
5.5.1 Simulated experts and test cases . . . 56
5.5.2 Baseline scenario . . . 57
5.5.3 Stage of implementation . . . 58
5.5.4 Expert dependency . . . 59
5.5.5 Expert heterogeneity . . . 59
5.5.6 Group size . . . 60
5.6 Conclusion . . . 61

6 Conclusion **63**
6.1 Main findings . . . 63
6.2 Contributions . . . 64
6.2.1 Contributions to the field of DSS . . . 64
6.2.2 Methodological contributions . . . 65
6.3 Discussion and recommendations for future research . . . 65
6.3.1 Formalizing the informative prior . . . 65
6.3.2 Methodology of Bayesian BAIT . . . 66
6.3.3 Methodology of the feedback loop simulations . . . 66
6.3.4 Investigated feedback loop effects . . . 67
6.4 Recommendations to Councyl . . . 68

A Overview of algorithms **75**
A.1 Formalizing the prior distribution . . . 75
A.2 Gibbs sampler for the probit model . . . 76
A.3 Metroplis-Hastings sampler for the logit model . . . 77
A.4 Variational Logistic Regression . . . 78
A.5 Data simulation . . . 79
A.6 Simulation of expert beliefs . . . 80

B Validation of Bayesian BAIT software **81**
B.1 Methodology . . . 81
B.2 Results . . . 82
B.2.1 Validation of the batch Gibbs sampler for the probit model . . . 82
B.2.2 Validation of the sequential Gibbs sampler for the probit model . . . 83
B.2.3 Validation of batch Metropolis-Hastings sampler for the logit model . . . 83
B.2.4 Validation of sequential Metropolis-Hastings sampler for the logit model . . . 84

C Plots of results Bayesian BAIT **85**
C.1 Markov chain convergence . . . 85
C.2 Comparison of posterior distributions . . . 92
C.3 Calibration plots . . . 95

D Simulated expert beliefs **97**
D.1 Regular experts . . . 97
D.2 Opposing experts . . . 98
D.3 Homogeneous experts . . . 98
D.4 Heterogeneous experts . . . 98

Chapter 1

Introduction

1.1 Decision Support Systems

For a long time, economists have been developing mathematical theories about how people make choices among a set of alternatives. This was stated by Ward Edwards, the father of behavioural decision theory, in his influential paper *The Theory of Decision Making* in 1954. Since Edwards developed his first theories on decision-making, the field of research has expanded enormously. A sea of approaches and methods have been developed to understand the behavioural process that leads to the choice an individual, organisation or expert makes (Train, 2009). Among other things, this knowledge expansion triggered a movement among researchers to study how decision theories can be applied to both understand and support decisions that are made in our society. This has led to the research area to which our research aims to make a contribution, which is the area of Decision Support Systems (DSS).

In general, two kinds of DSS have been developed: knowledge-based DSS and data-driven DSS (Montani & Striani, 2019). The knowledge-based DSS models human knowledge in computational terms and requires experts to self-report the rules they apply in their decision-making process. Although this system is well established, the rules in this system are often hard to determine and unable to grasp the fine trade-offs made in crucial decisions. The other group of DSS maintains a data-driven approach, as it applies machine learning methods to large historical data sets to abstract patterns that can then be used to make predictions. However, these models are particularly useful for very large amounts of data and often lack transparency in their internal logic, preventing experts to understand the root cause (Gretton, 2018).

Because of the inconveniences of these two approaches to DSS, Ten Broeke et al. (2021) have developed a new methodology that should find the golden mean: Behavioural Artificial Intelligence Technology (BAIT). BAIT is a DSS powered by discrete choice theory and AI techniques and is built around the notion that expertise is reflected in the decisions experts make. The objective of BAIT is to make accessible to a group of experts the combined expertise of their peers in the context of a particular decision problem. The technology codifies the expertise of this group of experts to yield two main insights. Firstly, BAIT explicates the impact of criteria for a particular decision task and secondly, the technology provides recommendations to experts for particular decision scenarios. As demonstrated by De Metz et al. (2021), BAIT is a feasible technique to gain insights into decision processes and has so far led to promising results.

1.2 Research goals

This paper studies two subjects related to BAIT. The first is related to the parameter estimation method behind the technology. Currently, BAIT is based on the classical parameter estimation method called Maximum Likelihood Estimation (MLE). The first part of the paper is dedicated to investigating a Bayesian approach as an alternative to MLE. The second subject concerns the potential effects of incorporating real-life decisions into BAIT, where our prime focus is the dynamics of the feedback loop. This subject has hitherto been unexplored and will be studied in the second part of the paper.

1.2.1 A Bayesian approach to BAIT

Bayesian inference is considered, because of the advantages of this statistical methodology and the shortcomings of MLE. In particular, Bayesian model assumptions match BAIT intrinsically better than classical estimation routines, which has three underlying reasons. Firstly, BAIT is looking for what it knows on the true impact of a variable on the decision task. Where MLE provides a parameter estimate based on the assumption that the data is a sample from the population, Bayesian analysis yields the belief about the true parameter, given the available data. Secondly, we desire BAIT to be able to incorporate prior expert knowledge on the impact of variables and to sequentially update parameters by new decisions being made. Bayesian inference is better equipped for fulfilling these desires than MLE, since it is able to include prior information into the inference procedure by assuming a prior distribution. Lastly, BAIT copes with small data sets and Bayesian approaches are proven to accommodate small data sets much better than classical approaches.

For BAIT, we are specifically interested in Bayesian versions of Generalized Linear Models (GLMs). One might wonder why we consider GLMs instead of powerful machine learning methods such as Random Forests or Neural Networks. This can be traced back to the emphasis of end-users of BAIT, that the technology should be designed in such a way that trust is maximized (Schrama, 2021). For the formation of initial trust, Ten Broeke et al. (2021) argue that BAIT should be interpretable and transparent, with a proper performance. Despite machine learning techniques typically having a good performance, these methods often lack transparency. GLMs have the advantage of providing an interpretable output, which offers transparency on the direct impact of variables on BAITs recommendation.

The first part of the paper investigates whether Bayesian inference would suit BAIT better than classical parameter estimation methods. In particular, we define several assessment criteria, design various Bayesian approaches and eventually assess the Bayesian approaches on these criteria. Hence, we have specified the following research goal for the first part of the paper:

1. Design several Bayesian alternatives to BAIT and assess which model is most suitable for the objectives of BAIT;

We find that Bayesian inference is a very appropriate alternative for the objectives of BAIT. Next to attractive model assumptions, this method of statistical inference offers a competitive predictive performance, well calibrated predictions and an elegant methodology for sequential updating of parameters. For future applications of BAIT, especially the deterministic posterior approximation method called Variational Logistic Regression (VLR) is recommended. Based on the first design of

Bayesian BAIT, we advise obtaining the prior density by applying MLE on the hypothetical data set. For future research, our paper provides several recommendations on how an informative prior for Bayesian BAIT could be formulated.

1.2.2 The feedback loop in BAIT

The increasing incorporation of artificial intelligence technologies within high impact decision ecosystems cause a growing need to understand the long-term behaviours of the decision systems and their potential consequences (D'Amour et al., 2020). Therefore, the second subject we focus on is related to the long-term consequences of implementing dynamic BAIT. Here, dynamics means that the model can be updated and improved every time a new real-life decision is being made. The largest concerns on the long-term effects are centred around the so-called "feedback loop": BAIT is updated with real-life choices, experts subsequently base their choices on the recommendation of BAIT and in the next step BAIT is updated with the decision of an expert that was already exposed to the recommendation of BAIT.

To gain a better understanding of the long-term consequences of employing a dynamic version of BAIT, the second part of the paper serves as an exploration of the consequences of the feedback loop. To the best of the author's knowledge, feedback loop dynamics have not yet been investigated in the field of dynamic expert DSS. Since dynamic BAIT is a prime example of a dynamic DSS, we believe that this study is a valuable contribution to this field of research. This brings us to the formulation of the research goal for the second part of the paper:

2. Explore the consequences of the feedback loop of dynamic BAIT and evaluate how these consequences are impacted by altering feedback loop conditions;

In our paper, we design a framework for simulating the feedback loop of dynamic BAIT. Related studies and the conceptualization of BAITs feedback loop point out that amplification of BAITs recommendations and homogenization of expert behaviour are two likely effects of continuous iterations through the feedback loop. Our simulations support the occurrence of these effects. Moreover, the effect of conditions such as the implementation stage of BAIT, expert dependency on BAIT, expert heterogeneity and the number of experts are explored. Studying these conditions indicates that recommendation amplification is intensified by assuming that expert dependency increases when recommendations become more extreme or by automating decisions from a threshold value. Homogenization of expert behaviour seems more likely to occur when BAITs recommendations are fully automated or in the case of one opponent in the group of experts.

1.3 Research structure

The remainder of this paper consists of five chapters. As a starter, Chapter 2 gives a more detailed explanation of the technology that we call BAIT. This is followed by Chapter 3, which describes the data that is used in our research. Chapter 4 then elaborately discusses our methods, results and conclusions on a Bayesian alternative to BAIT. Subsequently, Chapter 5 explores the occurrence and consequences of the feedback loop in dynamic BAIT. The paper is finalized by our conclusion and discussion in Chapter 6.

Chapter 2

Behavioural Artificial Intelligence Technology

2.1 Background information

To contextualize the research, this section provides insight into where BAIT is mainly applied and how the technology works. BAIT was first proposed by Ten Broeke et al. (2021) and generally serves two goals. First, it should make implicit decisions of experts explicit, by making the decision process more transparent. Secondly, the insights on the decision-making process are used to provide advice to experts for future decision tasks.

The technology is particularly designed for modelling complex decisions that only a small group of experts is eligible to make. BAIT is aimed at situations where decisions with a high degree of complexity, uncertainty and time pressure need to be made. Moreover, experts find the technology mostly useful in situations where even in retrospect it is difficult to objectively conclude whether the decision that was made was optimal. Examples of such situations are decisions on which migrants to grant a residence permit, which applicants to supply with a welfare allowance or which COVID-19 patients to admit to the ICU.

Ten Broeke et al. (2021) explain in five steps how BAIT works:

Step 1: The expert decision is specified and factors are identified that are presumed to play a role in the decision-making process of an expert;

Step 2: The choice model is specified. It is decided which terms, e.g. (non)-linear weights or interaction effects, are included and which choice model type is applied;

Step 3: A choice experiment is designed, in which a group of experts is invited to make a series of hypothetical choices based on scenarios that mimic real decision situations. These scenarios have an efficient design, such that each choice generates a maximum amount of information;

Step 4: The observed choices are used to estimate the importance weights of all factors using maximum likelihood techniques;

Step 5: The results are presented back to the experts. Factor weights are visualized, showing how each factor contributes to the experts' decisions, and new synthetic choice scenarios are included to illustrate how an assessment is made.

2.2 Case study

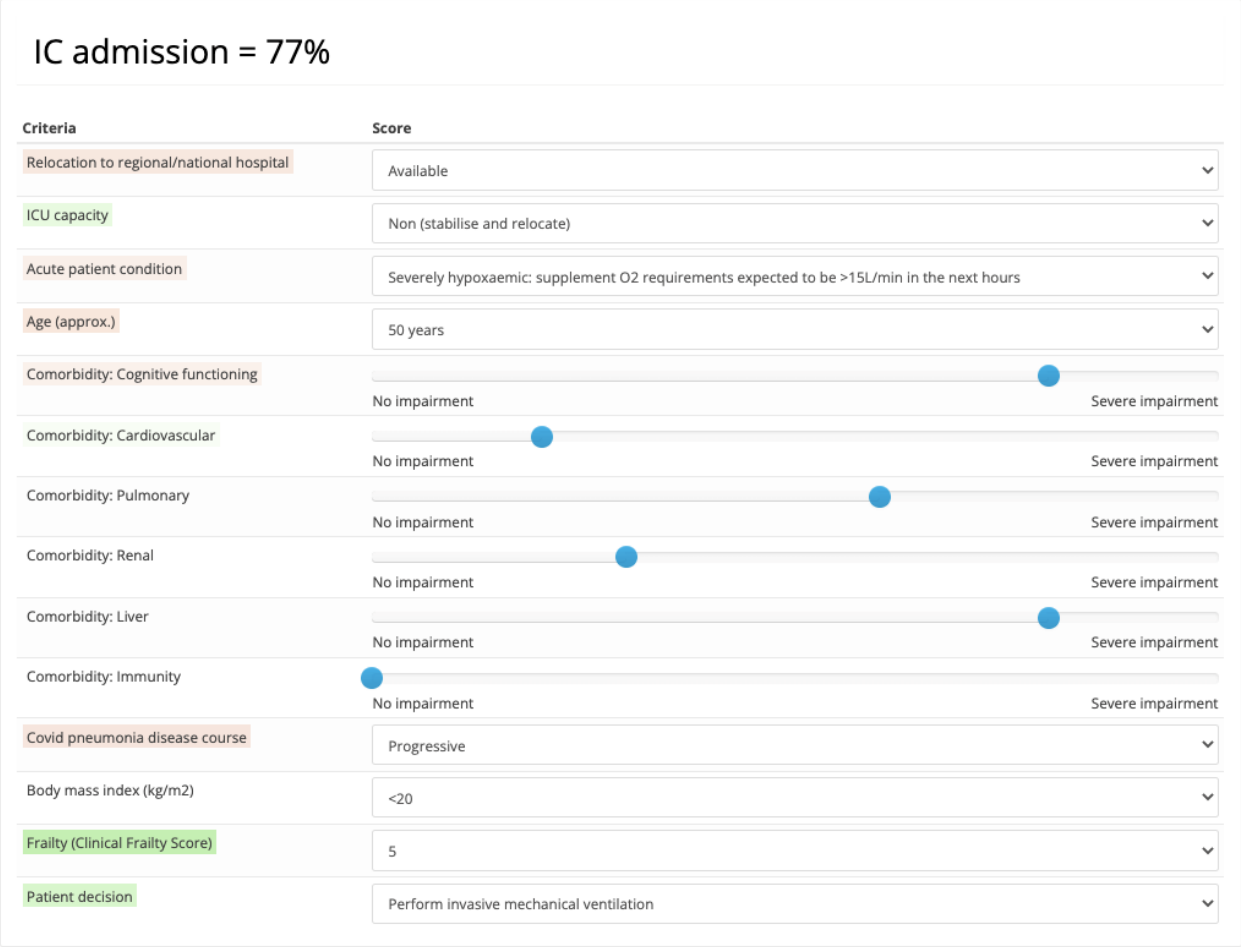
To gain a better understanding of the technology, we illustrate BAIT by a case study of De Metz et al. (2021). This case focuses on developing a model that explicates the implicit conditions Dutch intensivists use to determine ICU eligibility of COVID-19 patients. In Step 1, the researchers organised several brainstorm sessions with the intensivists, from which the factors that are used to evaluate the eligibility for ICU treatment of COVID-19 patients were specified. In Step 2, they used these inputs to specify a binary decision model with fourteen factors that are either binary or ordinal. These factors were then used in Step 3, where 25 hypothetical ICU admission scenarios were designed for maximum information content. Using a web application, these scenarios were presented to seventeen senior intensivists as well as fellows from the OLVG hospital. In Step 4, logistic regression was applied to obtain a model with the weights that intensivists implicitly assign to the various factors. In the final step, the results were presented back to the experts.

The model is presented in a technological environment such as in Figure 2.1. On the left side of this figure, we see the factors that were included in the model. The colour coding refers to the effect the factors have. Green means a positive effect, red a negative effect and no colour means no significant effect on the decision task. The brighter the colour, the larger the effect of that factor. In this environment, experts can provide the inputs of a new scenario they encounter. These inputs are then used to calculate the probability of a decision to be made, which in the context of BAIT is interpreted as the percentage of colleagues that would make a positive decision. For the case presented in Figure 2.1, we would interpret the model results as "77% of your colleagues would admit a patient with these specifications to the ICU".

2.3 Problem statement

The only source of information currently used for parameter estimation is the data from choice experiments. These choice experiments of hypothetical scenarios are designed in a single conjoint approach, which provides information on the trade-offs being made between attributes (Armstrong, 2001). However, there are two other data sources that contain information for parameter estimation: prior beliefs of experts and real-life decisions. The prior beliefs are collected during the set-up of the experiment, where experts explicate their intuition on the importance and sign of parameters. Moreover, experts make real-life decisions, which could also be included in the model to enhance its knowledge base. The latter was researched by Schrama (2021), who concluded that the inclusion of real-life decisions in BAIT would both be useful and feasible. For this reason, this paper studies how Bayesian approaches can include these two sources of information and what the long-term effects of including real-life decisions are.

Figure 2.1: Illustration of how the output of BAIT is presented to experts



Chapter 3

Data description

3.1 Introduction

As described in Chapter 2, we have access to three data sources for modelling BAIT. Firstly, the experts explicate their prior beliefs on the importance and sign of potential factors weighed in the decision-making process. Secondly, hypothetical choice scenarios provide a hypothetical input of variables to experts, who should then make a choice based on these variables. Thirdly, real-life choices made by experts are collected to update the model.

The data used in this research comes from TU Delft spin-off Councyl, which is the company that has developed and commercialized BAIT. The company provided two data sets. The primary data set concerns the decision made by experts of a Dutch municipality, who decide whether a household qualifies for a special welfare application. The second data set focuses on Dutch hospital Onze Lieve Vrouw Gasthuis (OLVG), where BAIT aims to assist intensivists in the decision which COVID-19 patients to admit to the ICU. In this chapter, we first describe both data sets and then discuss some data modifications we have applied.

3.2 Special welfare applications

3.2.1 Data description

In order for a household to qualify for a special welfare application, their case should be both strictly necessary and due to special circumstances. This paper only focuses on the choice of whether a special welfare application is strictly necessary. The aim of modelling this decision task is to explicate the implicit criteria experts of the municipality maintain when making this decision. The factors used for evaluating the eligibility of households were first determined by Councyl and experts together. Subsequently, choice experiments containing 30 hypothetical welfare application scenarios were presented to eight experts. Therefore, the experimental data set contains 240 observations. The decisions made for these scenarios can be found in Table 3.1. Although the experts received exactly identical scenarios, there was no unanimity in the decisions for more than half of the decision tasks. This illustrates how subjective these decision tasks can be and that experts are not convinced of the existence of a ground truth for every scenario.

In our Bayesian approach to BAIT, we do not only review the hypothetical decisions, but also

real-life decisions being made. Therefore, this research includes 67 real-life decisions concerning 44 unique cases. The cases have been reviewed by a minimum of one and a maximum of five experts. All of these experts also participated in the choice experiments. The results can be found in Table 3.2. In total 33 decisions are positive and 34 are negative, which means the real-life data set is balanced.

Table 3.1: Decisions made in the experimental special welfare data set

Scenario ID	29905	29906	29907	29908	29909	29910	29911	29912	29913	29914	29915	29916	29917	29918	29919
No special circumstances		8	8	7	8	3	6	6	5	5	4	8	7	8	6
Special circumstances	8			1		5	2	2	3	3	4		1		2
Scenario ID	29920	29921	29922	29923	29924	29925	29926	29927	29928	29929	29930	29931	29932	29933	29934
No special circumstances	6	6	5	8	4	8	8	6	8	7	7	8	7	8	6
Special circumstances	2	2	3		4			2		1	1		1		2

Table 3.2: Decisions made in the real special welfare data set

Scenario ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	16	17	18	19	20	21	22	23
No special circumstances				1	1		1	1												2		
Special circumstances	2	3	3	2	1	1			1	1	1	1	1	1	1	1	1	1	1		1	1
Scenario ID	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
No special circumstances			4	4	3	2	3					1	1	1	2	1	1	1	1	1	1	1
Special circumstances	1	1		1		1		1	1	1	1											

3.2.2 Variable description

In collaboration with the experts from the municipality, Councyl determined six key criteria whether a special welfare application is labelled as strictly necessary. Due to confidentiality between Councyl and the municipality, the exact names of these variables should remain anonymous. For this reason, we label the variables *Var 1* to *Var 6*. Nevertheless, we provide descriptions of the variables in a randomized order: life circumstances of a household; strictly urgent reasons of the household to apply for a special welfare allowance; fact-based proof of necessity; costs made by the household in the past that could have been prevented; family situation; whether the special welfare application would serve to purchase a first life necessity.

The prior beliefs on the characteristics can be found in Table 3.3. For each characteristic, the experts give an indication of the expected importance, type and sign. For importance they can choose four ascending levels: "nice to have", "important", "critical" or "knock-out". The latter level requires special attention. Knock-out variables entail that when a certain value for this variable is met, experts are entirely sure a particular decision should be made. Since there is no uncertainty in the effect of knock-outs, it does not hold any statistical value. Therefore, we have chosen to leave knock-out variables and observations which attain a knock-out level out of this research. *Var 6* is such a variable, which is thus not included in the research.

The sign (or shape) refers to the belief of the expert whether the impact of the characteristic on his final decision will be positive, negative or *inverted u-shape* (the latter only applies to ordinal variables with non-linear levels). Furthermore, the data type describes how the characteristics should be accounted for in the model. All ordinal variables have exactly three levels and can be included in the model in three different ways. When a variable is stated to be ordinal with linear levels, they are included with one single parameter. Alternatively, when a variable is ordinal with

non-linear levels, we dummy encode the variables. This dummy encoding transforms the levels into binary variables. Each level is then compared to a reference level, which is chosen to be the most frequently occurring level in the real-life data set. Lastly, variables can be ordinal with linear levels, between which can be interpolated. These variables are included as continuous variables.

Descriptive statistics of the experimental and real-life data set are presented in Table 3.4. Overall, we see that in the experimental data the attributes were distributed evenly to obtain the maximum information, but real-life decisions contain more variation.

Table 3.3: Prior beliefs on the special welfare characteristics

Characteristic	Importance	Shape/sign	Type
Var 1	Critical	Positive	Ordinal with non-linear levels
Var 2	Nice to have	Positive	Ordinal with non-linear levels
Var 3	Important	Positive	Binary
Var 4	Critical	Positive	Binary
Var 5	Important	Positive	Ordinal with linear levels and interpolation
Var 6	Knock-out	Positive	Binary

Table 3.4: Background characteristics of the experimental and real-life special welfare data sets

				Experimental data		Real-life data	
Characteristics	Attribute	Min	Max	Mean	St. Dev	Mean	St. Dev
Var 1	Level 0	0	1	0.367	0.483	0.087	0.284
	Level 1	0	1	0.300	0.459	0.087	0.284
	Level 2	0	1	0.333	0.472	0.826	0.382
Var 2	Level 0	0	1	0.367	0.483	0.507	0.504
	Level 1	0	1	0.300	0.459	0.101	0.304
	Level 2	0	1	0.333	0.472	0.391	0.492
	Var 3	0	1	0.500	0.501	0.638	0.484
	Var 4	0	1	0.500	0.501	0.725	0.450
	Var 5	0	2	0.967	0.838	1.276	0.706
	Var 6	0	1	NA	NA	0.029	0.169
	Expert decision	0	1	0.204	0.404	0.507	0.504

3.3 COVID-19 ICU uptakes

3.3.1 Data description

In the second application of Bayesian BAIT, we make use of the data set provided by Dutch hospital Onze Lieve Vrouwe Gasthuis (OLVG). This is the same data set as used in the case study in Section 2.2. The data was collected by De Metz et al. (2021), who used it for an illustration of the performance of classical BAIT. The aim of their research was to develop a model that explicates the implicit conditions Dutch intensivists use to determine ICU eligibility of COVID-19 patients. The factors used for evaluating the eligibility of COVID-19 patients were first determined by the researchers and intensivists together. Subsequently, choice experiments containing 25 hypothetical ICU admission scenarios were presented to 13 intensivists and four fellows of the OLVG. Therefore the experimental data contains 425 observations. The decisions made for these scenarios can be found in Table 3.5a. Also in this data set, we see there is not a lot of consensus between experts, as the decisions are only unanimous for one scenario.

Whereas De Metz et al. (2021) only include data from the choice experiments, we also include 70 real-life decisions concerning 23 patients. The patients have been reviewed by a minimum of one and a maximum of four experts. The decisions can be found in Table 3.5b. We should note that there are some minor variations in the input of scenarios that were reviewed more than once. This is due to subjective views of experts or sensitivity tests applied by Councyl. The real-life data set is imbalanced, since 54 decisions were positive and only 16 were negative.

Table 3.5: Decisions made in the experimental and real-life COVID-19 data sets

(a) Decisions in the experimental data set

Scenario ID	Don't admit	Admit
1416	16	1
1417	0	17
1418	5	12
1419	7	10
1420	7	10
1421	15	2
1422	6	11
1423	6	11
1424	2	15
1425	7	10
1426	5	12
1427	7	10
1428	3	14
1429	1	16
1430	1	16
1431	8	9
1432	6	11
1433	3	14
1434	2	15
1435	9	8
1436	1	16
1437	6	11
1438	2	15
1439	5	12
1440	4	13

(b) Decisions in the real-life data set

Scenario ID	Don't admit	Admit
1	0	4
2	0	4
3	0	4
4	1	4
5	0	4
6	3	0
7	2	1
8	0	4
9	1	3
10	3	0
11	2	1
12	0	3
13	0	3
14	0	3
15	0	3
16	3	0
17	0	3
18	0	3
19	0	2
20	0	2
21	0	1
22	1	1
23	0	1

3.3.2 Variable description

In collaboration with the experts from the OLVG, De Metz et al. (2021) have determined fourteen key criteria for deciding whether to admit a COVID-19 patient to the ICU. The prior beliefs on these characteristics can be found in Table 3.6. Descriptive statistics of the experimental and real-life data set are presented in Table 3.7.

ICU Capacity, Acute, BMI and *Frailty* are all ordinal variables with three possible inputs. *ICU Capacity* indicates the available capacity on the ICU, which can be none, limited or spacious. ICU capacity is believed to be of critical and positive influence on the expert decision. *Acute* describes how acute the threat of the patient is, where "not hypoxic, but progressive" is the least acute and "acute respiratory threat" is the most. Acuteness is thought to be critical and to increase the probability of an ICU uptake. *BMI* is short for Body Mass Index and is an approximate measure of whether a patient is overweight or underweight. Below 20 a patient is qualified as underweight, whereas over 40 suggests morbid obesity. Both these extremes are believed to reduce the probability of ICU uptake. The characteristic *frailty* originates from the international renown 9-point Clinical Frailty Score, which summarizes a patient's overall level of fitness or frailty. For the purposes of BAIT, this frailty score is reduced to a 3-points scale. Experts believe that more fragile patients

Table 3.6: Prior beliefs on COVID-19 characteristics

Characteristic	Importance	Shape/sign	Type
ICU Capacity	Critical	Positive	Ordinal with linear levels
Acute patient condition	Critical	Positive	Ordinal with linear levels
BMI	Nice to have	Inverted u-shape	Ordinal with non-linear levels
Frailty	Critical	Positive	Ordinal with linear levels
Relocation to regional/national hospitals	Nice to have	Positive	Binary
Age	Critical	Negative	Ordinal with linear levels and interpolation
Comorbidity: Cognitive functioning	Critical	Negative	Ordinal with linear levels and interpolation
Comorbidity: Cardiovascular	Critical	Negative	Ordinal with linear levels and interpolation
Comorbidity: Pulmonary	Critical	Negative	Ordinal with linear levels and interpolation
Comorbidity: Renal	Critical	Negative	Ordinal with linear levels and interpolation
Comorbidity: Liver	Critical	Negative	Ordinal with linear levels and interpolation
Comorbidity: Immunity	Important	Negative	Ordinal with linear levels and interpolation
COVID pneumonia disease course	Nice to have	Negative	Binary
Patient preference	Important	Positive	Binary

Table 3.7: Background characteristics of the experimental and real-life COVID-19 data sets

				Experimental data		real-life data	
Characteristics	Attribute	Min	Max	Mean	St. Dev	Mean	St. Dev
ICU Capacity	None	0	1	0.360	0.481	0.000	0.000
	Limited	0	1	0.320	0.467	0.100	0.302
	Spacious	0	1	0.320	0.467	0.900	0.302
Acute	Acute resparatory threat	0	1	0.360	0.481	0.129	0.337
	Significantly hypoxic	0	1	0.320	0.467	0.586	0.496
	Not hypoxic, but progressive	0	1	0.320	0.467	0.286	0.455
BMI	<20	0	1	0.320	0.467	0.014	0.120
	20 - 40	0	1	0.360	0.481	0.971	0.168
	>40	0	1	0.320	0.467	0.014	0.120
Frailty	Daily medical assistence	0	1	0.400	0.490	0.157	0.367
	Medical issues, but independent	0	1	0.320	0.467	0.143	0.352
	Fit	0	1	0.280	0.450	0.700	0.462
	Relocation possibilities	0	1	0.480	0.500	1.000	0.000
	Age	30	70	50.800	16.492	58.743	12.039
	Comorbidity: Cognitive functioning	0	2	0.840	0.834	0.021	0.078
	Comorbidity: Cardiovascular	0	2	1.000	0.850	0.269	0.514
	Comorbidity: Pulmonary	0	2	0.960	0.774	0.064	0.154
	Comorbidity: Renal	0	2	0.880	0.766	0.361	0.638
	Comorbidity: Liver	0	2	0.720	0.827	0.009	0.053
	Comorbidity: Immunity	0	2	0.840	0.785	0.090	0.362
	COVID pneumonia disease course	0	1	0.480	0.500	0.737	0.438
	Patient decision	0	1	0.480	0.500	0.957	0.204
	Expert decision	0	1	0.685	0.465	0.771	0.423

have a lower chance of admittance to the ICU.

Furthermore, there are three binary variables in this data set. *Relocation possibilities* state whether there is a possibility to relocate patients to other hospitals if necessary, in which case it is believed that patients are more easily accepted to the ICU. For all patients in the real-life data set this possibility exists. *COVID pneumonia disease course* states the progress of the disease, where 0 is said to be progressive and 1 means COVID with complications. Overall, experts primarily believe that more complications lead to a lower chance of ICU uptake, albeit more frequently appearing. *Patient preference* implies whether a patient wishes to undergo invasive ventilation, which would be necessary if a patient at the ICU would get respiratory problems. The descriptive statistics show that most patients wish to undergo such a treatment.

Finally, several variables are ordinal with interpolation, which means that they are included as

continuous. Such a variable is *age*, which can range between 30 and 70. Patients in the real-life data set that are younger than 30 or older than 70 are restricted to these minimum/maximum ages, which can be seen as a limitation on the data. Experts believe younger patients are sooner admitted to the ICU. Lastly, the experts have indicated that *comorbidities*, such as cognitive functioning or cardiovascular, are also deemed to have a critical impact on ICU uptake. A higher score on such comorbidities on the scale 0 (normal), 1 (mild) or 2 (strongly restricted) is believed to have a negative effect on ICU uptake.

3.4 Data modifications

Two main data modifications have been applied to both data sets to make them better suitable for modelling BAIT. Firstly, the continuous variables and ordinal variables with linear levels are scaled to a $[0, 1]$ scale using *min-max scaling*. In general, Bayesian analysis is insensitive to scale transformations of individual independent variables, given the priors are changed accordingly (Raftery, 1996). However, applying an automatic default standardization procedure could offer a way to substantially improve the understanding of the output of a model (Gelman, 2008). In Bayesian BAIT, min-max scaling comes in particularly useful as the non-binary variables are then presented on the same scale as the binary and dummy-encoded variables. Formally the scaling is presented as:

$$\tilde{x}_{ij} = \frac{x_{ij} - min(\mathbf{x_j})}{max(\mathbf{x_j}) - min(\mathbf{x_j})},\ i = 1, \ldots, N. \tag{3.1}$$

Here x_{ij} denotes observation i of variable vector $\mathbf{x_j}$ and $min(\mathbf{x_j})$ and $max(\mathbf{x_j})$ denote the respective minimum and maximum value of $\mathbf{x_j}$.

Secondly, the missing entries of independent variables in the real-life data set have been imputed. Where the special welfare data set contains 11 missing entries, the COVID-19 data set contains 25. Those entries are either unavailable due to not being asked by the expert or they are actually unknown. We propose to impute these missing entries by a sequential imputation method. First, entries are attempted to be imputed by the entry value of an observation with the same scenario ID. Since scenarios have been judged by various experts, one scenario often has multiple observations (see Tables 3.2 and 3.5b). These other observations are used to impute the one with the missing observation. This has resulted in the imputation of respectively 7 (out of 11) and 23 (out of 25) entries in the special welfare and COVID-19 data sets. Hereafter, k-Nearest Neighbours (kNN) imputation has been applied to impute the remaining missing entries. This method is known for its simplicity, easy understanding and relatively high accuracy (Zhang, 2012). The kNN imputation is designed to find k nearest neighbours for missing entries from all complete entries in a given dataset. The missing entry is imputed by the most frequently occurring neighbour if the target feature is categorical or with the mean of the neighbours if the target feature is numerical. We have chosen k to be 10, considering that the method is relatively insensitive to the exact value of k within the range of 10 to 20 neighbours (Troyanskaya et al., 2001) and that our real-life data set is small.

Chapter 4

A Bayesian approach to BAIT

4.1 Introduction

The main focus of Chapter 4 is developing a Bayesian approach to BAIT. In particular, we have formulated the following research goal:

1. Design several Bayesian alternatives to BAIT and assess which model is most suitable for the objectives of BAIT;

In order to assess the multiple models, five assessment criteria have been determined with respect to the objectives of BAIT. All assessment criteria are considered important for a proper functioning of BAIT and the satisfaction of end-users. These criteria are used at the end of this chapter to come to a conclusion to which approach to BAIT can best be maintained.

(i) *Matching model assumptions.* The first criterion is that the model assumptions properly match the objectives and applications of BAIT. The Literature Review below provides a comparison of the model assumptions.

(ii) *Suitable for sequential updating.* A strong desire for BAIT is that the technology should be able to sequentially update the initial model estimation by iteratively feeding real-life observations. The empirical research evaluates this possibility for every Bayesian approach.

(iii) *Good predictive performance.* A successful implementation of any AI technology cannot go without a good predictive performance. Therefore, it is crucial to take into account how well BAIT predicts future decisions, as this gives an indication of the quality of BAITs recommendation. Several performance metrics are employed to test the predictive performance.

(iv) *Good calibration of predicted probabilities.* We want the recommendation of BAIT to be in line with the true probability that an expert would make a decision. For example, if a model is well calibrated and predicts 0.8 for a number of similar decision tasks, about 80% of these decision tasks should prove to be positive.

(v) *Short computation time.* When experts are actively using BAIT, it is important that recommendations can be generated in real-time. In a situation where an expert is under time pressure, the computation time of a model should not form an obstacle.

This chapter is kicked off by two sections of literature reviews. In Section 4.2 we first review the literature on the background of Bayesian inference and on the motivation why this kind of inference is applicable for the case of BAIT. This is followed by a literature review on designing a Bayesian methodology for BAIT in Section 4.3. Section 4.4 then explains the Bayesian methods we have designed to model BAIT. This is followed by a comparison of the results of these methodologies for our two respective data sets in Sections 4.5 and 4.6. Finally, Section 4.7 provides a conclusion on which approach to BAIT can best be maintained based on our assessment criteria.

4.2 Literature review: Background and motivation

This section serves as background and motivation for the use of Bayesian inference in BAIT. Section 4.2.1 provides an introduction to Bayesian econometrics. This is followed by Section 4.2.2, which focuses on the application of Bayesian methods in Decision Support Software (DSS). The section is concluded by Section 4.2.3, which provides a motivation why we consider applying Bayesian methods to BAIT.

4.2.1 Introduction to Bayesian inference

As an opposing view to regularly applied statistical methods (i.e. frequentist methods) Bayesian methods have become increasingly popular. The introduction of Bayesian techniques dates back to reverend Thomas Bayes (1701 - 1761). His philosophy was seldom used until the early 1960s when applications of the Bayesian viewpoint to econometric models started to gain more attention (Greenberg, 2012). In these days, however, computational intensity and the lack of statistical software imposed difficulties for researchers willing to apply Bayesian analysis. Due to recent advances in technology and the introduction of software packages, this kind of analysis is nowadays gaining in popularity (Albers et al., 2018).

Before diving into Bayesian methods, it might be wise to define where we use Bayesian inference. The general goal of statistical inference is to learn about the values of parameters in a model. Since there is uncertainty on these values, Bayesian inference is built on the idea that the parameter is a random variable with a probability distribution. Before seeing the data, the parameter is assigned a *prior* distribution. Bayesian inference centres on finding the *posterior* distribution, the distribution of the parameter conditional on having observed the data. The fundamental result of Bayesian inference can be formulated as "the posterior distribution of a parameter is proportional to the likelihood function times the prior distribution" (Greenberg, 2012). Important to note is that Bayesian inference refers to an estimation method and not to a behavioural model. Probit, logit, mixed logit, or any other model can be estimated by either frequentist or Bayesian procedures (Train, 2009).

4.2.2 Bayesian inference in DSS

To the best of the author's knowledge, the usage of Bayesian inference in Decision Support Systems (DSS) remains relatively unexplored[1]. The most comparable study found was conducted by Kim

[1]Bayesian inference should not be confused with Bayesian Networks, which is a widely used methodology for decision-makers to improve their decision-making

et al. (2014). This study first specified a multi-attribute utility function (MAUF) to model the subjective utility of diverse decision-makers, which could be interesting for Bayesian BAIT in case one wants to model taste heterogeneity between experts. Moreover, the authors have applied Bayesian inference to obtain the posterior distribution of unknown quantities in MAUF, such as the expected utility and weighting factors. Their Bayesian application however deviates significantly from ours, as they make use of a continuous dependent variable, have chosen for an uninformative prior distribution and have straightforwardly implemented a software program to perform their Bayesian model. These methods are too far away from our desired approach to directly adopt them. Nevertheless, this study shows that Bayesian inference is able to provide meaningful information to decision-makers to improve their decision-making process.

4.2.3 Bayesian inference in BAIT

When reviewing the literature, Bayesian inference seems to be much more aligned with the methodology of BAIT than the existing frequentist estimation techniques. This has three main reasons. The first of those reasons is that the interpretation of parameter estimates in Bayesian inference seems to match the reasoning behind BAIT better. Classical MLE assumes that input data is a random sample from a larger population. The standard deviations in a frequentist model reflect how parameter estimates might vary over repeated applications of random sampling from the population. This contradicts with Bayesian inference, which assumes that parameters are random variables themselves. Bayesian analysis yields a belief about the true parameter, given the available data (Jackman, 2009). The latter perspective is better suitable for BAIT, as it provides end-users insights into the distribution of true parameters and the standard deviation provides insight on the uncertainty about those parameters.

Secondly, Bayesian analysis is appropriate for BAIT, since it allows to incorporate *a priori* knowledge of experts and offers a more elegant design for processing real-life decisions. The Bayesian methodology offers the possibility to make prior beliefs explicit and to moderate these prior beliefs by actual data at hand (Kaplan & Depaoli, 2013). Robert et al. (2007) argue that Bayesian methods therefore much closer mimic the actual decision-making process than classical approaches. The prior distribution is able to summarize the available information (or even the lack of information) about the parameters at hand. The posterior distribution then follows from a confrontation between priors and experiments. Moreover, Spiegelhalter et al. (1993) have reviewed the use of Bayesian analysis in expert systems and acknowledged that when having access to real-life cases, Bayesian statistical methods are very useful in updating the original subjective input in priors.

Thirdly, BAIT makes use of small-sized expert data and it is crucial to point out that Bayesian methods are generally preferred to frequentist methods in case of small data sets. Daziano & Bolduc (2013) argue that frequentist methods are mostly based on asymptotic properties, which is why these can suffer from small sample bias. Contrarily, Bayesian inference is statistically exact and thus works independently of the sample size. Nevertheless, Bayesian estimates are highly sensitive to the specification of the prior distribution. If this aspect is not managed, Bayesian estimates can actually be worse than frequentist methods (McNeish, 2016). However, Daziano & Bolduc (2013) found that when the sample size is low and the prior is properly specified, Bayesian point estimates outperform maximum likelihood in terms of accuracy and efficiency and provide tighter confidence intervals. Lee & Song (2004) go as far as stating that Bayesian methods will produce accurate

parameter estimates and reliable goodness-of-fit tests when the sample sizes equal only four or five times the number of parameters. In their simulation study, they found that parameter estimates of Bayesian approaches are much closer to the true value than the parameter estimates of MLE approaches. There are numerous studies that also reached the conclusion that Bayesian methods better accommodate reduced sample sizes (Wanless et al., 2015; Doron & Gaudreau, 2014; Kliem et al., 2010; Stenling et al., 2015; MacNab et al., 2011).

4.3 Literature review: Bayesian methodology

This section serves as a literature study on the methodology that we use for designing several Bayesian approaches to BAIT. Section 4.3.1 starts by introducing behavioural models for binary decision tasks. Section 4.3.2 studies the considerations when defining a prior. This is followed by Section 4.3.3, which contextualizes two important terms for this paper, which are batch and sequential estimation. Eventually, Sections 4.3.4 to 4.3.6 elaborate on different methodologies for approximating the posterior density.

4.3.1 Binary behavioural models

For the simplest version of BAIT, the choice task is binary and the data is cross-sectional. The most frequently applied behavioural models for such decision models are *logit* and *probit*. Both models allow to analyze the effects of a set of independent variables on a binary dependent variable (Walsh, 1987). What differentiates the two models is the assumed distribution of the error term: for logit, this is the logistic distribution and for probit the normal distribution.

The current model in static BAIT is based on a binary logit model, as this is the easiest and most widely used discrete choice model. Logit's popularity is due to the fact that the formula for the choice probabilities takes a closed-form and is readily interpretable. Nevertheless, logit does reveal issues when moving to a multinomial setting, as this imposes the Independence of Irrelevant Alternatives (IIA) assumption (Train, 2009). This assumption can be alleviated by switching to a probit model. However, the issue of a probit model in a multinomial setting is that it does not take on a closed-form expression. This may cause computational difficulties when the number of categories increases. A problem of both models is that the distributional assumptions of the error term impose restrictions on the distribution of unobserved components of utilities.

The plainest forms of both models do not allow for heterogeneity among experts, although extensions exist[2]. Nevertheless, heterogeneity among experts is not included in our design of Bayesian BAIT, since this is not in line with the goal of the technology. BAIT is not intended to provide insights on varying preferences among experts, as this would amplify segregation in experts opinions.

4.3.2 Prior distributions

To obtain the posterior distribution in a Bayesian model, one should first have specified the prior distribution. The choice of prior distribution is a key issue in Bayesian analysis. Most formally the prior serves to encode information germane to the problem being analyzed. However, in practice it often becomes a means of stabilizing inferences in complex, high-dimensional problems (Gelman et

[2]See McFadden & Train (2000) for the mixed logit model and Hausman & Wise (1978) for the probit extension.

al., 2017). For the prior, a researcher should account for the choice of the distribution and the degree of information included. The prior distribution can be specified either *proper* or *improper*. We say a prior distribution is improper when the distribution integrates to infinity and proper when the distribution integrates to unity. A proper prior can be specified for a parameter when its possible values are concentrated between certain bounds. The main reason why a proper prior is preferred is because it is crucial for the use of Bayes' factors and comparing models (Greenberg, 2012).

Moreover, the amount of information that is incorporated in the prior is subject to discussion. Prior distributions that include a researcher's prior information are labelled as *informative*. Proponents of informative priors argue that not including prior knowledge represents ignorance (Bernardo & Smith, 2009). However, opponents find the obvious dependence of the posterior results on informative priors somewhat disturbing (Bernardo, 1979). This is because using an informative prior cannot go without the introduction of some extent of subjective judgement (Kass & Wasserman, 1996). The opponents of informative priors are therefore in favour of *non-informative priors*, which do not include prior information.

In any case, Lee & Song (2004) argue that one should rather use non-informative prior inputs than bad informative prior inputs. In the context of small samples, such as BAIT, McNeish (2016) mentions that the importance of selecting the prior distribution cannot be underestimated. The author refers to several methodological studies that have shown that MCMC methods can outperform frequentist methods in small-sample contexts. However, these studies obtained improved performance by specifying informative priors based on expert opinions or previous studies. Gelman (2006) also provides a warning that in case the data supply gets really small, a noninformative prior distribution can lead to a posterior distribution that is improper or is proper but unrealistically broad; both of which are undesirable. For all these reasons, we attempt to specify a prior for BAIT that is both proper and informative, where we select our prior inputs with great care.

4.3.3 Batch and sequential estimation

In the process of developing a Bayesian approach to BAIT, the distinction between batch and sequential estimation is crucial to make. Let us first define these two estimation methods. A model is batch estimated when the model processes all data points at once. In sequential estimation, data points are considered one at a time, and the model parameters are updated after each such presentation (Bishop, 2006). The model is thus updated with respect to the previous iteration of the model.

One of the downsides of classical BAIT is that it only allows batch estimation of the model. However, when BAIT would be applied in a dynamic setting, real-life decisions would arrive in a continuous stream and predictions must be made before new decisions arrive. Bishop (2006) argues that in such a setting, sequential learning is appropriate and that Bayesian methods are intrinsically well suited to sequential learning.

The advantage of sequential learning is that at every iteration BAITs belief is slightly altered. Note that as the number of observations increases the contribution from successive data points gets smaller (Bishop, 2006). However, there is the risk of running into an accumulation of approximation errors in a sequential learning scheme. Posterior distributions never exactly follow the assumed type of distribution. This imposes a minor error when approximating the posterior by this distribution. During the course of the sequential estimation procedure, there is the risk that these approximation

errors add up substantially (Jaakkola & Jordan, 2000). We will explore the use of sequential estimation in Bayesian BAIT and evaluate its approximation errors by comparing the posterior distributions of sequential and batch estimation.

4.3.4 Markov Chain Monte Carlo methods

Since the models that- we consider do not allow for direct computation of posterior distributions, we need to resort to approximation schemes. Generally, there exist two main classes for these approximation schemes, which are either stochastic or deterministic (Bishop, 2006). Stochastic approximation techniques are most commonly applied and are reviewed in this section. Subsequently, deterministic approximation schemes are considered in Section 4.3.6. The most widely utilized group of stochastic approximation techniques are Markov Chain Monte Carlo (MCMC) methods. MCMC methods simulate a Markov chain whose equilibrium distribution is, under some regularity conditions, the posterior density of interest (Tanner, 2012). MCMC methods are labelled as approximate, since the number of simulations on which the parameters of the posterior distribution are based is finite.

Chib & Greenberg (1996) have made an elaborate study on MCMC methods and argue that combined with data augmentation these methods can be used to organize a systematic approach to Bayesian inference. The authors describe the two main techniques for sampling from the posterior distribution: the Metropolis-Hasting algorithm and Gibbs sampling. In the Metropolis-Hastings (MH) algorithm, initiated by Metropolis et al. (1953) and later extended by Hastings (1970), the next value in a Markov chain is generated from a proposal density and then accepted or rejected according to the target density at the candidate point relative to the density at the current point. The alternative to the MH algorithm is the Gibbs sampler, which was proposed by the brothers Geman & Geman (1984) and further developed by Tanner & Wong (1987) and Gelfand & Smith (1990). The Gibbs sampler can be seen as a special case of the MH algorithm. This method prescribes partitioning the random vector of parameters into several blocks and defining the transition density as the product set of full conditional densities. The next item in the Markov chain is then obtained by successively sampling the full conditional densities, given the most recent values of the conditioning parameters.

A downside to the usage of MCMC methods is a possible slow convergence of the Markov Chain. Slow convergence could occur due to (a combination of) three root causes. Primarily, convergence is slow when the various parameters in the Markov chain are highly correlated *a posteriori* (Chib & Carlin, 1999). Secondly, the "tallness" (i.e. the number of individual data points) of the data set can be a problem. Matrix multiplication with large matrices can be computationally intensive. Data tallness is particularly a problem in the MH algorithm when examining whether a proposed distribution is accepted. The algorithm then needs to sweep over the whole data set, at each and every iteration, for the evaluation of the likelihood function (Robert et al., 2018). However, since the expert data sets used in BAIT are small-sized, this problem is likely to be mitigated. Thirdly, the efficiency of the MH algorithm depends crucially on the scaling of the proposal density (Gelman et al., 1997). If the variance of the proposed density is too small, the Markov Chain will converge slowly since all its increments will be small. Conversely, when the variance is too large, the chain will reject a too large proportion of its proposed moves. Therefore, the choice of such a proposal density should be closely monitored in the Bayesian application of BAIT.

4.3.5 MCMC methods for binary models

To be able to apply MCMC methods, we review which behavioural model requires the use of which particular MCMC method. In the extensive study of Train (2009) on discrete choice modelling, the author states that Gibbs sampling should be used "when it is difficult to draw directly from the joint density and yet easy to draw from the conditional density of each element given the values of the other elements". Contrarily, the MH algorithm should only be applied as a sampling technique "if all else fails", since it is applicable to practically any distribution. Such a situation could be when the posterior distribution for one parameter conditional on the other parameters does not take a simple form (Chib & Greenberg, 1995).

The initial choice task of Bayesian BAIT reviewed in this paper is binary, for which two behavioural models are considered: logit and probit. Simulation from the posterior is complicated by the fact that no conjugate prior exists for the parameters in both models, without the introduction of data augmentation (Held & Holmes, 2006). When the priors are non-conjugate, the conditional posteriors are not of a known distribution and direct sampling is not possible. For these cases, Gibbs sampling cannot be applied. Therefore, Albert & Chib (1993) have proposed a straightforward data augmentation approach that provides a general framework for analyzing binary regression models. Through this approach, the conditional distributions of the model parameters are equivalent to those of the Bayesian normal linear regression model with Gaussian noise. Hence, conditional conjugate priors are available to the conditional likelihood and the Gibbs sampler can be applied.

Nevertheless, such a straightforward data augmentation approach for applying the Gibbs sampler for the logit model does not exist in the current literature. Held & Holmes (2006) have proposed a method to apply Gibbs sampling with the introduction of several auxiliary variables. Nevertheless, since these auxiliary variables have no trivial interpretation, we believe implementing this model would lead to a great loss of interpretability to an already complex algorithm. As an alternative, the MH algorithm can be used for models that are not conditionally conjugate (Gelman et al., 1995). Therefore, we will apply the augmented MH algorithm for the logit model, which is described by Frühwirth-Schnatter & Frühwirth (2010). This approach also follows the data augmentation approach of Albert & Chib (1993).

4.3.6 Variational Logistic Regression

As stated in Section 4.3.4, there exist two classes of approximation schemes: stochastic and deterministic. Since the class of stochastic approximation techniques (i.e. MCMC) have clear disadvantages, we also consider the class of deterministic approximation schemes for BAIT. Variational approximation procedures are an example of a deterministic approach. The main idea behind a variational method is to convert a complex problem into a simpler problem by including variational parameters (Jordan et al., 1999).

Jaakkola & Jordan (2000) first introduced the concept of variational approximation methods for Bayesian logit models, which we refer to as Variational Logistic Regression (VLR). VLR involves finding a variational transformation of the logit model and using this transformation as an approximation to the likelihood. When we assume a Gaussian prior, the goal of VLR is to approximate the posterior as good as possible within the class of Gaussian distributions.

Jaakkola & Jordan (2000) found that VLR can be exploited to yield closed-form expressions that approximate the posterior distributions for the parameters in logistic regression. This leads to a much faster computation of a posterior distribution than the simulation methods in the MCMC chain. Moreover, an important feature of this variational approach is that it is well suited for both sequential and batch learning (Bishop, 2006). This matches the desire of BAIT to incorporate sequential updating of parameters. Nevertheless, a disadvantage is that VLR only generates approximations to the posterior distribution, such that it never leads to exact results (Bishop, 2006).

4.4 Methodology

The methodology section is kicked off by outlining the various approaches to BAIT that are designed in this paper. In Section 4.4.2 follows a discussion on the methodology for modelling binary models, after which we elaborate on the difference between MLE and Bayesian approaches. Subsequently, Section 4.4.4 provides a description of how we obtain our prior specification. Then follow Sections 4.4.5 to 4.4.7 in which the three Bayesian approaches designed for BAIT are respectively described. The methodology section is concluded in Section 4.4.8 by discussing our validation methods and performance metrics.

4.4.1 Outlining the various approaches

We apply various approaches to yield parameter estimates. In total, we model eleven approaches, that are being compared and contrasted in this paper. An overview of these approaches can be found in Table 4.1. The discussion is centred around four main points of discussion:

- *MLE vs. Bayesian.* The first discussion is whether a frequentist or Bayesian approach should be maintained. For this purpose, we compare three different settings. The first setting is a classical Maximum Likelihood Estimation (MLE). The second is a partially Bayesian approach, where the prior is obtained by applying MLE on the experimental data, after which Bayesian methods are applied on the real-life decisions to obtain the posterior. The third setting is a fully Bayesian approach, where the prior is based on the experts' prior beliefs and Bayesian methods are performed on the entire experimental and real-life data set together.

- *Sequential vs. batch estimation.* The second discussion is around sequential or batch estimation. MLE only allows for batch estimation, but for Bayesian analysis, both can be applied. A sequential Bayesian approach to BAIT is preferred, but might suffer from an accumulation of approximation errors. Therefore, we compare a batch Bayesian approach to a "batch - sequential" Bayesian approach in the fully Bayesian setting. Here, a batch - sequential approach means that batch estimation is used on the experimental data and sequential estimation on the real-life decisions. The batch Bayesian approach serves as a benchmark to evaluate whether the parameters of the batch - sequential approaches do not drift too far away.

- *Logit vs. probit.* The initial behavioural model of binary BAIT is logit. However, since a logit model does not necessarily outperform a probit model, we compare the two approaches. In the frequentist setting, MLE can be applied for both logit and probit. In a Bayesian setting, these behavioural models call for different methods.

- *Bayesian method.* Three Bayesian methods are compared. As stochastic approximation approaches, two Markov Chain Monte Carlo (MCMC) methods are applied. For the probit and logit model, we respectively make use of the Gibbs sampler and Metropolis-Hastings (MH) sampler. Variational Logistic Regression (VLR) is applied to explore the use of a deterministic approximation alternative.

Table 4.1: Summary of the various approaches

Name model	MLE vs. Bayesian	Sequential vs. Batch	Logit vs. probit	Bayesian method
MLE probit	MLE	Batch	Probit	-
MLE-based sequential Gibbs	Partially Bayesian	Batch - sequential	Probit	Gibbs
Batch Gibbs	Fully Bayesian	Batch	Probit	Gibbs
Batch - sequential Gibbs	Fully Bayesian	Batch - sequential	Probit	Gibbs
MLE logit	MLE	Batch	Logit	-
MLE-based sequential MH	Partially Bayesian	Batch - sequential	Logit	MH
Batch MH	Fully Bayesian	Batch	Logit	MH
Batch - sequential MH	Fully Bayesian	Batch - sequential	Logit	MH
MLE-based sequential VLR	Partially Bayesian	Batch - sequential	Logit	VLR
Batch VLR	Fully Bayesian	Batch	Logit	VLR
Batch - sequential VLR	Fully Bayesian	Batch - sequential	Logit	VLR

4.4.2 Binary models

Let us introduce data matrix X, which consists of N observations of p independent variables. Moreover, $\boldsymbol{y}$ is the vector of the N corresponding binary choices. In BAIT we are looking for the vector of effects $\boldsymbol{\beta}$ of independent variables X on the dependent variable $\boldsymbol{y}$. The two models reviewed for obtaining these effects are logit and probit. In the following, both models are explained based on the work of Train (2009).

The logit model is based on binary choice task i, where an expert is faced with two alternatives: $Y_i = 1$ or $Y_i = 0$. Each alternative has its own utility function. Since there are only two alternatives, the utility of alternative 0 is scaled to 0. By definition of the utility function, the expert makes decision $Y_i = 1$ if the corresponding utility function $U_i > 0$ and decides $Y_i = 0$ otherwise. We make the assumption that U_i is linear in its parameters:

$$U_i^L = \boldsymbol{x}_i' \boldsymbol{\beta}^L + \epsilon_i^L \tag{4.1}$$

In the logit model, the error term ϵ_i^L is assumed to follow a logistic distribution, which results in the following probability for a positive decision:

$$P^L[Y_i = 1 | \boldsymbol{\beta}^L] = \frac{e^{\boldsymbol{x}_i' \boldsymbol{\beta}^L}}{1 + e^{\boldsymbol{x}_i' \boldsymbol{\beta}^L}} \tag{4.2}$$

The probit model shows many similarities to the logit model. The model also scales the utility of alternative 0 to 0 and is centred around modelling utility U_i of alternative 1. Expert i makes decision $Y_i = 1$ if $U_i > 0$ and $Y_i = 0$ otherwise. The utility function for the probit model is given by:

$$U_i^P = \boldsymbol{x}_i' \boldsymbol{\beta}^P + \epsilon_i^P \tag{4.3}$$

The probit model is built around the assumption that the error terms ϵ_i^P are independently and identically normally distributed around mean zero. Let us denote $\Phi(\cdot)$ as the cumulative density function of the standard normal distribution. The probability of a positive decision is obtained by:

$$P^P[Y_i = 1|\boldsymbol{\beta}^P] = \Phi(\boldsymbol{x}_i'\boldsymbol{\beta}^P) \tag{4.4}$$

Irrespective of whether the concerning model is probit or logit, the likelihood function $p(\boldsymbol{y}|\boldsymbol{\beta})$ is taken as the joint probability of decision vector $\boldsymbol{y}$, given parameter values $\boldsymbol{\beta}$. This can be written as:

$$p(\boldsymbol{y}|\boldsymbol{\beta}) = \prod_{i=1}^{N} P[Y_i = y_i|\boldsymbol{\beta}] \tag{4.5}$$

4.4.3 MLE and Bayesian approaches

In this section, we elaborate on the fundamental difference between MLE and Bayesian approaches. In MLE, we yield estimated parameter vector $\hat{\boldsymbol{\beta}}$ by finding the parameters that maximize the likelihood function provided in Equation 4.5. When maintaining the MLE approach, $\hat{\boldsymbol{\beta}}$ is interpreted as a vector of deterministic parameters estimates.

A Bayesian model distinguishes itself by assuming the effects of the independent variables follow their own distribution. Before seeing the data, we can state our prior beliefs on the distribution of these effects, which we capture in prior density $p(\boldsymbol{\beta})$. After reviewing the data, we can use the prior distribution and likelihood of the data to arrive at posterior distribution $p(\boldsymbol{\beta}|\boldsymbol{y})$. This posterior distribution summarizes our beliefs about the effect of the independent variables, given the prior distribution and the data. This is formulated in Bayes' theorem as follows:

$$p(\boldsymbol{\beta}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{\beta})p(\boldsymbol{\beta})}{p(\boldsymbol{y})} \propto p(\boldsymbol{y}|\boldsymbol{\beta})p(\boldsymbol{\beta}) \tag{4.6}$$

Bayes' theorem describes the methodology for a batch Bayesian approach. For a sequential Bayesian approach, where every observation is added iteratively, the theorem can be slightly altered:

$$\begin{aligned} p(\boldsymbol{\beta}|y_1, \ldots, y_N) &= \frac{p(y_1, \ldots, y_N|\boldsymbol{\beta})p(\boldsymbol{\beta})}{p(y_1, \ldots, y_N)} \\ &= \frac{p(y_N|\boldsymbol{\beta})p(y_1, \ldots, y_{N-1}|\boldsymbol{\beta})p(\boldsymbol{\beta})}{p(y_N)p(y_1, \ldots, y_{N-1})} \\ &= \frac{p(y_N|\boldsymbol{\beta})p(\boldsymbol{\beta}|y_1, \ldots, y_{N-1})}{p(y_N)} \\ &\propto p(y_N|\boldsymbol{\beta})p(\boldsymbol{\beta}|y_1, \ldots, y_{N-1}) \end{aligned} \tag{4.7}$$

In this sequential approach, the posterior after having observed data points $y_1, \ldots, y_{N-1}$ serves as prior when data point y_N is absorbed. This happens at every iteration. Several possibilities exist for yielding the intermediate posteriors. We choose to approximate posteriors of the previous iteration by a Gaussian distribution, since these provide conditionally conjugate priors. Nevertheless, this approach might suffer from an approximation error, because the posterior distributions do not necessarily follow a Gaussian distribution.

4.4.4 Formalizing the prior distribution

For the fully Bayesian models, the prior distribution captures the prior beliefs on the parameters as explicated by the experts. We use both binary logit and probit as behavioural models. As discussed in the Section 4.3, we should aim for a proper prior, which integral can be computed. For the probit model, a Gaussian prior would be a logical choice, as this prior is conditionally conjugate when applying the data augmented Markov Chain proposed by Albert & Chib (1993). For the logit model, no (conditional) conjugate priors exist, without the introduction of several untrivial auxiliary variables. Therefore, we also assume a Gaussian prior density for the logit model. Assuming the same form of the prior distribution as in the probit case, allows us to effectively compare the logit and probit models. Moreover, a Gaussian distribution is the most common choice of prior in a logit model in the literature (Fussl et al., 2013; Held & Holmes, 2006; Raftery, 1996; Frühwirth-Schnatter & Frühwirth, 2010). The Gaussian prior distribution is defined as:

$$p(\boldsymbol{\beta}) \sim N(\tilde{\boldsymbol{\beta}}, \tilde{\boldsymbol{B}}) \tag{4.8}$$

Prior parameters $\tilde{\boldsymbol{\beta}}$ and $\tilde{\boldsymbol{B}}$ are determined by two inputs provided by experts: the expected marginal effects (EMEs) of the independent variables and the expected base probability of a positive decision. A marginal effect (ME) tells us how a unit change in a certain independent variable changes the dependent variable, i.e. the decision an expert makes. EMEs should be obtained in the design phase of the experiment, where experts explicate the ME they expect variables $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p$ to have on their decision. These EMEs are then be used to compute $\tilde{\boldsymbol{\beta}}$. Next to the EMEs, experts should also state a 95% confidence interval for the MEs. These are used to compute $\tilde{\boldsymbol{B}}$.

Furthermore, we also require the input of an expected base probability of a positive decision and the 95% confidence interval of this probability. The base probability is denoted by P_{base} and is defined as the expected average probability that an individual expert makes a positive decision, i.e. $Y_i = 1$. From P_{base} we can determine the expected base utility U_{base}, which is used for two purposes. Firstly, the MEs for either a logit or a probit model are not a linear function of the utility. Therefore, the MEs should be evaluated for a base scenario. Secondly, U_{base} is used for the calibration of the prior expectation of the intercept parameter, $\tilde{\beta}_0$. The confidence interval of P_{base} is then used to obtain the prior standard deviation of the intercept parameter, $\tilde{\sigma}_0$.

As the probit and logit model are based on different underlying densities, their corresponding parameters $\boldsymbol{\beta}$ are also scaled differently[3]. However, to specify the parameters of the prior density, we apply the same general methodology. Let us first define the probability density functions (PDF) and cumulative density functions (CDF) of the respective models. The logit model is based on the logistic model, which PDF g and CDF G are given by:

$$g(\mathbf{x_i}'\boldsymbol{\beta}) = \frac{exp(\mathbf{x_i}'\boldsymbol{\beta})}{(1 + exp(\mathbf{x_i}'\boldsymbol{\beta}))^2} \tag{4.9}$$

$$G(\mathbf{x_i}'\boldsymbol{\beta}) = \frac{exp(\mathbf{x_i}'\boldsymbol{\beta})}{1 + exp(\mathbf{x_i}'\boldsymbol{\beta})} \tag{4.10}$$

[3]As the parameters are scaled differently, Amemiya (1981) proposed an approximate scaling factor between the parameters of the logit and probit model: $\beta^L \approx 1.6\beta^P$. We use this scaling factor to approximately verify whether our resulting parameter estimates are scaled correctly.

On the other hand, the probit model is based on a standard Gaussian distribution, which PDF and CDF are respectively denoted by ϕ and Φ:

$$\phi(\mathbf{x_i}'\boldsymbol{\beta}) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(\mathbf{x_i}'\boldsymbol{\beta}^2)} \tag{4.11}$$

$$\Phi(\mathbf{x_i}'\boldsymbol{\beta}) = \int_{-\infty}^{\mathbf{x_i}'\boldsymbol{\beta}} \phi(z)dz \tag{4.12}$$

Now, let us define a general distribution that can be either logistic or standard normal. We denote its PDF by h and CDF by H, such that $h \in \{g, \phi\}$ and $H \in \{G, \Phi\}$. Using this general distribution, we are able to derive the prior parameters for both models.

We start by defining the probability of decision being positive: $P[Y_i = 1] = H(\mathbf{x_i}'\boldsymbol{\beta})$. Using this formula, we can obtain base utility $U^L_{base} = H^{-1}(P_{base})$. Moreover, let us denote prior location parameter vector by $\tilde{\boldsymbol{\beta}}$. We derive this vector by means of the EME of the variables. First, we should state the difference between continuous and dummy variables, as this is important for the derivation of the EMEs. We make use of variables which are either "binary", "ordinal with linear levels", "ordinal with linear levels with interpolation", or "dummy-encoded ordinal with nonlinear levels". From these possible variable types, we consider "ordinal with linear levels with interpolation" to be continuous and all the others to behave like dummies. The formulas for the ME of continuous and dummy variables are given by:

$$\text{ME}_{j,cont} = \frac{\partial P[Y_i = 1|U_i = U_{base}]}{\partial \boldsymbol{x}_{ij}} = h(U_{base})\beta_j \tag{4.13}$$

$$\begin{aligned} \text{ME}_{j,dum} &= P[Y_i = 1|U_i = U_{base} + 1 * \beta_j] - P[Y_i = 1|U_i = U_{base} + 0 * \beta_j] \\ &= H(U_{base} + \beta_j) - H(U_{base}) \end{aligned} \tag{4.14}$$

Taking the inverse of these formulas, we get the following formulas for the location parameters of the prior distribution forthe decision model:

$$\tilde{\beta}_{j,cont} = \frac{\text{EME}_j}{h(U_{base})} \tag{4.15}$$

$$\tilde{\beta}_{j,dum} = H^{-1}\left(\text{EME}_j + H(U_{base})\right) - U_{base} \tag{4.16}$$

Next, we focus on setting prior covariance matrix $\tilde{\boldsymbol{B}}$. We assume that $\boldsymbol{\beta}$ are distributed independently a priori, as this corresponds to the situation where the individual variables are of interest in their own right. Hence, $\tilde{\boldsymbol{B}}$ can be seen as a diagonal matrix with on the diagional variance parameters $\tilde{\sigma}^2_0, \tilde{\sigma}^2_1, \ldots, \tilde{\sigma}^2_p$. We set these variance parameters according to the confidence interval of the EMEs as stated by the experts. This confidence interval can be denoted as $(\text{LBEME}_j, \text{UBEME}_j)$. Here LBEME_j is the lower bound of the confidence interval and UBEME_j the upper bound. For parameter estimates in a regression context, the lower and upper bounds of the confidence interval are typically defined as $lb[\beta_j] = \tilde{\beta}_j - \phi^{-1}[1 - \frac{\alpha}{2}] * \tilde{\sigma}_j$ and $ub[\beta_j] = \tilde{\beta}_j + \phi^{-1}[1 - \frac{\alpha}{2}] * \tilde{\sigma}_j$. Since we have assumed a 95% confidence interval, $\phi^{-1}[1 - \frac{\alpha}{2}] \approx 1.96$.

For yielding an expression for $\tilde{\sigma}_j$ we make use of LBEME_j. UBEME_j is not used, but would have resulted in the same findings. The formulas for computing LBEME_j for respectively continuous and

dummy variables are stated as follows:

$$\text{LBEME}_{j,cont} = \frac{\partial P[Y_i = 1 | U_i = U_{base}, \beta_j = \tilde{\beta}_j - 1.96\tilde{\sigma}_j]}{\partial \boldsymbol{x}_{ij}} = h(U_{base})(\tilde{\beta}_j - 1.96\tilde{\sigma}_j) \tag{4.17}$$

$$\begin{aligned} \text{LBEME}_{j,dum} &= P[Y_i = 1 | U_i = U_{base} + 1 * \beta_j, \beta_j = \tilde{\beta}_j - 1.96\tilde{\sigma}_j] \\ &\quad - P[Y_i = 1 | U_i = U_{base} + 0 * \beta_j, \beta_j = \tilde{\beta}_j - 1.96\tilde{\sigma}_j] \\ &= H(U_{base} + \tilde{\beta}_j - 1.96\tilde{\sigma}_j) - H(U_{base}) \end{aligned} \tag{4.18}$$

Solving these formulas for $\tilde{\sigma}_j$, we get:

$$\tilde{\sigma}_{j,cont} = \frac{1}{1.96}\left(\tilde{\beta}_j - \frac{\text{LBEME}_j}{h(U_{base})}\right) \tag{4.19}$$

$$\tilde{\sigma}_{j,dum} = \frac{1}{1.96}\left(U_{base} + \tilde{\beta}_j - H^{-1}\left(LBEME_j + H(U_{base})\right)\right) \tag{4.20}$$

The only parameters left to determine are $\tilde{\beta}_0$ and $\tilde{\sigma}_0$. The intercept term represents the expected mean utility, given that all $x = 0$. Similar to all other prior parameters, the prior parameters of the intercept term should also be set according to the base scenario. We have first set all parameters individually in accordance with their EME. This has provided us with hyperparameter vector $\tilde{\boldsymbol{\beta}}_{1:p} = [\tilde{\beta}_1, \ldots, \tilde{\beta}_p]$. Since we desire the prior utility to be calibrated around U_{base}, intercept term β_0 should serve as a calibration point around U_{base} with respect to the remaining part of the utility function U_{rem}. Formally, this reads as:

$$U_{base} = \beta_0 + U_{rem} \tag{4.21}$$

To obtain prior intercept parameter $\tilde{\beta}_0$, we rewrite this equation as:

$$\beta_0 = U_{base} - U_{rem} \tag{4.22}$$

From this equation, we know U_{base}, but still have to determine remaining utility U_{rem}. We estimate U_{rem} by $\tilde{U}_{rem} = \overline{\boldsymbol{x}}'_{1:p}\tilde{\boldsymbol{\beta}}_{1:p}$, where $\overline{\boldsymbol{x}}_{1:p}$ is the vector of means over independent variables $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p$. However, in this stage of the research we have not observed the data yet. Therefore, we define proxy mean data vector $\tilde{\boldsymbol{x}}_{1:p}$, which holds a proxy for the mean of variables $1, \ldots, p$. Since we have no prior intuition on the distribution of the data, we set all proxy means to the average of the interval on which they are defined. Taking into account the different type of variables that we have, we retrieve proxy mean value $\tilde{x}_j$ for variable j by:

$$\tilde{x}_j = \begin{cases} \frac{1}{N_{ord}} & \text{if variable } j \text{ is a dummy-encoded ordinal variable with } N_{ord} \text{ levels} \\ \frac{a+b}{2} & \text{if variable } j \text{ is not dummy-encoded and is defined on the interval [a,b]}^4 \end{cases} \tag{4.23}$$

Using this proxy vector, we get the following expression for $\tilde{\beta}_0$:

$$\tilde{\beta}_0 = U_{base} - \tilde{U}_{rem} = U_{base} - \tilde{\boldsymbol{x}}'_{1:p}\tilde{\boldsymbol{\beta}}_{1:p} \tag{4.24}$$

[4]Since we apply min-max scaling to interval $[0, 1]$, $\tilde{x}_j = \frac{1}{2}$ for all not dummy-encoded variables.

Secondly, we should obtain an expression for σ_0, which represents the uncertainty in β_0. We base the uncertainty in β_0 on the uncertainty in U_{base}, which originates from the uncertainty in P_{base}. For this reason, we set σ_0 according to the 95% confidence interval of P_{base}. This confidence interval can be denoted by $(\text{LBP}_{base}, \text{UBP}_{base})$, where LBP_{base} is the lower bound of the confidence interval and UBP_{base} the upper bound. Similar to before, we base our expression for σ_0 on LBP_{base}, but using UBP_{base} would have led to the same expression. We know that the lower bound of the 95% confidence interval for β_0 is given by $lb[\beta_0] = \tilde{\beta}_0 - 1.96 * \tilde{\sigma}_0$. Based on the definitions of P_{base} and U_{base}, we get:

$$\begin{aligned} LBP_{base} &= H(lb[U_{base}]) \\ &= H(lb[\beta_0] + \tilde{U}_{rem}) \\ &= H(\tilde{\beta}_0 - 1.96 * \tilde{\sigma}_0 + \tilde{U}^L_{rem}) \\ &= H(U_{base} - 1.96 * \tilde{\sigma}_0) \end{aligned} \tag{4.25}$$

This leaves us with the following expression for $\tilde{\sigma}_0$:

$$\tilde{\sigma}_0 = \frac{1}{1.96}\left(U_{base} - H^{-1}(LBP_{base})\right) \tag{4.26}$$

The complete methodology for obtaining the prior specification is summarized by Algorithm 1 in Appendix A.1. Respectively replacing h and H by logistic density functions g and G results in the prior parameters for the logit model, and replacing them by standard normal density functions ϕ and Φ allows us to obtain the prior parameters for the probit model.

4.4.5 Gibbs sampler for the probit model

The first considered behavioural model for the binary decision task is the probit model. For this model, we have assumed a Gaussian prior distribution. Albert & Chib (1993) have introduced an elegant methodology for applying Gibbs sampling for the probit model using data augmentation. Since this methodology remains one of the most popular methods for Bayesian analysis for binary choice tasks, we also follow this approach.

We start our methodology by introducing variable vector $\boldsymbol{Z} = (Z_1, \ldots, Z_N)$, where Z_i independently follow normal distribution $N(\boldsymbol{x}_i^T\boldsymbol{\beta}, 1)$. Furthermore, we define $Y_i = 1$ if $Z_i > 0$ and $Y_i = 0$ otherwise. This definition is similar to the definition of utility function U_i^P and can be interpreted likewise. The joint posterior density of $\boldsymbol{\beta}$ and $\boldsymbol{z}$, given data vector $\boldsymbol{y}$, is given by:

$$p(\boldsymbol{\beta}, \boldsymbol{z}|\boldsymbol{y}) \propto p(\boldsymbol{\beta}) \prod_{i=1}^{N} \left(\mathbf{1}\{z_i > 0\}\mathbf{1}\{Y_i = 1\} + \mathbf{1}\{z_i \leq 0\}\mathbf{1}\{Y_i = 0\}\right) \phi(z_i; \boldsymbol{x}_i^T\boldsymbol{\beta}, 1) \tag{4.27}$$

From this, we obtain an expression for the posterior density of $\boldsymbol{\beta}$, given $\boldsymbol{z}$ and $\boldsymbol{y}$:

$$p(\boldsymbol{\beta}|\boldsymbol{z}, \boldsymbol{y}) \propto p(\boldsymbol{\beta}) \prod_{i=1}^{N} \phi(z_i; \boldsymbol{x}_i^T\boldsymbol{\beta}, 1) \tag{4.28}$$

In Equation 4.8, the Gaussian prior was defined as $P^p(\boldsymbol{\beta}) \sim N(\tilde{\boldsymbol{\beta}}^P, \tilde{\boldsymbol{B}}^P)$. Albert & Chib (1993)

then show that the conditional posterior mean $\hat{\boldsymbol{\beta}}^P$ and covariance matrix $\hat{\boldsymbol{B}}^P$ are given by:

$$\hat{\boldsymbol{\beta}}^P = (\hat{\boldsymbol{B}}^P)^{-1}\left((\tilde{\boldsymbol{B}}^P)^{-1}\tilde{\boldsymbol{\beta}}^P + \boldsymbol{X}'\boldsymbol{z}\right) \tag{4.29}$$

$$\hat{\boldsymbol{B}}^P = \left((\tilde{\boldsymbol{B}}^P)^{-1} + \boldsymbol{X}'\boldsymbol{X}\right)^{-1} \tag{4.30}$$

Next, note from Equation 4.27 that the posterior distribution of $\boldsymbol{Z}$, conditional on $\boldsymbol{\beta}$, also has a simple form. The random variables $\boldsymbol{Z} = (Z_1, \ldots, Z_N)$ are independent with conditional distribution function described by:

$$Z_i|\boldsymbol{y}, \boldsymbol{\beta} \sim \begin{cases} N(\boldsymbol{x}_i'\boldsymbol{\beta}, 1) \text{ truncated at the left by } 0 & \text{if } y_i = 1 \\ N(\boldsymbol{x}_i'\boldsymbol{\beta}, 1) \text{ truncated at the right by } 0 & \text{if } y_i = 0 \end{cases} \tag{4.31}$$

With respect to these mathematical definitions, we can draw up the sampling scheme for the Gibbs sampler. The Gibbs sampler simulates dependent draws that are approximately from the probability distribution of interest $p(\boldsymbol{\beta}, \boldsymbol{z}|\boldsymbol{y})$. In each simulation step of the Gibbs sampler, we sample $p(\boldsymbol{\beta}|\boldsymbol{z}, \boldsymbol{y})$ and $p(z_i|\boldsymbol{\beta}, \boldsymbol{y})$.

In order to use the Gibbs sampler, we should determine how long the simulation needs to be run. In these simulations, we take both "burn-in simulations" and "thinning" into account. Burn-in simulations are essential, since the arbitrary starting point of our chain is unlikely to directly come from the targeted stationary posterior distribution. Thinning means only selecting every nth observation of the Markov chain and is useful to decorrelate the Markov chain. While burn-ins and thinning increase the computation time, they reduce the number of simulations ultimately saved from a run from the Gibbs sampler (Raftery & Lewis, 1995). Since we consider burn-ins and thinning absolutely necessary, we discard N_{burn} initial simulations from the Markov chain and take thinning value n_{thin}. The number of stored simulations is denoted by N_{sim}, which brings the total number of simulations to $N_{burn} + N_{sim} \cdot n_{thin}$. The complete sampling scheme for the Gibbs sampler is given in Algorithm 2 in Appendix A.2.

4.4.6 Metropolis-Hastings sampler for the logit model

The second possibility for the behavioural model of binary Bayesian BAIT is the logit model, for which we apply the Metropolis-Hastings (MH) sampler as MCMC method. For the implementation of the MH algorithm for the binary logit model, we follow the difference random utility model as described by Frühwirth-Schnatter & Frühwirth (2010). Their methodology is chosen, since their data augmentation approach leads to a conditionally Gaussian model, which allows efficient sampling of the random variables. Interesting to note is that their model is based on the works of Albert & Chib (1993), which we previously used in the Gibbs sampler for the probit model.

First we review a general introduction to the MH algorithm based on the work Chib & Greenberg (1995). The algorithm is built around finding target density $p(\boldsymbol{\beta}|\boldsymbol{y})$. To generate samples from $p(\boldsymbol{\beta}|\boldsymbol{y})$, we need to find a transition kernel $T(\boldsymbol{\beta}^*|\boldsymbol{\beta})$ whose nth iterate converges to $p(\boldsymbol{\beta}|\boldsymbol{y})$ for large n. The process is started at arbitrary value $\boldsymbol{\beta}^{(0)}$ and iterated a large number of times. After this large number of iterations, the distribution of the observations generated from the simulation should approximate the target distribution.

The issue is to find an appropriate $T(\boldsymbol{\beta}^*|\boldsymbol{\beta}^{(m)})$. The right transition kernel will ultimately

lead us to the target density. Therefore, we propose a candidate-generating density $q(\boldsymbol{\beta}^*|\boldsymbol{\beta}^{(m)})$. This density is defined such that $\int q(\boldsymbol{\beta}^*|\boldsymbol{\beta}^{(m)})d\boldsymbol{\beta}^* = 1$ and can be interpreted as saying that when a Markov chain is at point $\boldsymbol{\beta}^{(m)}$, the density generates a value $\boldsymbol{\beta}^*$ from $q(\boldsymbol{\beta}^*|\boldsymbol{\beta}^{(m)})$. Since it is unlikely that every $\boldsymbol{\beta}^*$ matches the target density, we introduce $\alpha(\boldsymbol{\beta}^*|\boldsymbol{\beta}^{(m)})$, which indicates the likeliness of moving from $\boldsymbol{\beta}^{(m)}$ to $\boldsymbol{\beta}^*$. If the move is not made, the process returns $\boldsymbol{\beta}^{(m)}$ as a value from the target distribution. The transition kernel is then defined as $T(\boldsymbol{\beta}^*|\boldsymbol{\beta}) = q(\boldsymbol{\beta}^*|\boldsymbol{\beta}^{(m)})\alpha(\boldsymbol{\beta}^*|\boldsymbol{\beta}^{(m)})$.

This introduction provides the foundation for the MH algorithm for the binary logit model, which we consider next. Frühwirth-Schnatter & Frühwirth (2010) propose to use an independence chain, which means that candidates $\boldsymbol{\beta}^*$ are drawn independent of current location $\boldsymbol{\beta}^{(m)}$. We consider the independence chain to be suitable for our sequential application of the MH algorithm, where data points are absorbed one at a time. This means that the candidate-generating density is likely to be close to the posterior density. The independence chain is stated to perform well when the posterior distribution is not too different in shape from the candidate-generating density (Tierney, 1994).

Similar to the work of Albert & Chib (1993), we introduce auxiliary variable vector $\boldsymbol{Z} = (Z_1, \ldots, Z_N)$. This vector translates to $Y_i = 1$ if $Z_i > 0$ and $Y_i = 0$ otherwise, which means that the interpretation of Z_i is similar to utility function U_i^L. In this scheme, we iteratively draw values from distributions $p(\boldsymbol{z}|\boldsymbol{\beta}^{(m)}, \boldsymbol{y})$ and $p(\boldsymbol{\beta}|\boldsymbol{z}^{(m)}, \boldsymbol{y})$. The formal definition of Z_i is given by:

$$Z_i = \boldsymbol{x}_i'\boldsymbol{\beta} + \epsilon_i, \quad \text{where } \epsilon_i \sim \text{Logistic}(0) \tag{4.32}$$

As was defined in Equation 4.9, we denote $g(\cdot)$ as the PDF of the logistic function. This results in the conditional likelihood function $p(\boldsymbol{z}|\boldsymbol{\beta})$ being written as:

$$p(\boldsymbol{z}|\boldsymbol{\beta}) = \prod_{i=1}^{N} g(z_i - \boldsymbol{x}_i'\boldsymbol{\beta}) \tag{4.33}$$

Let us remind that the parameters of the prior distribution are given by location vector $\boldsymbol{b}^{(0)} = \tilde{\boldsymbol{\beta}}^L$ and covariance matrix $\boldsymbol{B}^{(0)} = \tilde{\boldsymbol{B}}^L$. Location vector $\boldsymbol{b}^{(0)}$ is used as starting value for $\boldsymbol{\beta}$ in the Markov chain. We specify candidate-generating density $q(\boldsymbol{\beta}^*|\boldsymbol{z})$ according to an independence chain, where the candidates are drawn independently of the current location $\boldsymbol{\beta}^{(m)}$. Proposal density $q(\boldsymbol{\beta}^*|\boldsymbol{z})$ is constructed by approximating the logistic distribution of the error term of Z_i by a normal distribution with mean zero and variance $\frac{\pi^2}{3}$. Candidate vector $\boldsymbol{\beta}^*$ is then generated from proposal density $q(\boldsymbol{\beta}^*|\boldsymbol{z}) = N(\boldsymbol{b}^{(m)}, \hat{\boldsymbol{B}})$, where location parameter $\boldsymbol{b}^{(m)}$ and variance parameter $\hat{\boldsymbol{B}}$ are defined as:

$$\boldsymbol{b}^{(m)} = \hat{\boldsymbol{B}}\left((\boldsymbol{B}^{(0)})^{-1}\boldsymbol{b}^{(0)} + \frac{3}{\pi^2}\boldsymbol{X}'\boldsymbol{z}\right) \tag{4.34}$$

$$\hat{\boldsymbol{B}} = \left((\boldsymbol{B}^{(0)})^{-1} + \frac{3}{\pi^2}\boldsymbol{X}'\boldsymbol{X}\right)^{-1} \tag{4.35}$$

In this augmented MH algorithm the target density is defined as $p(\boldsymbol{\beta}|\boldsymbol{z})$ and is proportional to

$p(\boldsymbol{z}|\boldsymbol{\beta})p(\boldsymbol{\beta})$. The expression we yield for the probability of move is given by:

$$\alpha(\boldsymbol{\beta}^*|\boldsymbol{\beta}^{(m)}, \boldsymbol{z}) = \min\left[\frac{p(\boldsymbol{z}|\boldsymbol{\beta}^*)p(\boldsymbol{\beta}^*)q(\boldsymbol{\beta}^{(m)}|\boldsymbol{z})}{p(\boldsymbol{z}|\boldsymbol{\beta}^{(m)})p(\boldsymbol{\beta}^{(m)})q(\boldsymbol{\beta}^*|\boldsymbol{z})}, 1\right] \tag{4.36}$$

Now knowing the acceptance probability of $\boldsymbol{\beta}^*$, we draw a random variable u from a uniform distribution on the interval $[0, 1]$. We then have:

$$\boldsymbol{\beta}^{(m+1)} = \begin{cases} \boldsymbol{\beta}^* & \text{if } u < \alpha(\boldsymbol{\beta}^*|\boldsymbol{\beta}^{(m)}, \boldsymbol{z}) \\ \boldsymbol{\beta}^{(m)} & \text{otherwise} \end{cases} \tag{4.37}$$

The final operation we need to execute in step m of the MH-algorithm is drawing a new vector $\boldsymbol{z}$ from distribution $p(\boldsymbol{z}|\boldsymbol{\beta}^{(m)}, \boldsymbol{y})$. The marginal densities for the z_i's are independent truncated logistic distributions., given by:

$$z_i|\boldsymbol{\beta}, y_i \sim \begin{cases} G(\boldsymbol{x}_i'\boldsymbol{\beta})\mathbf{1}\{z_i > 0\} & \text{if } y_i = 1 \\ G(\boldsymbol{x}_i'\boldsymbol{\beta})\mathbf{1}\{z_i \leq 0\} & \text{if } y_i = 0 \end{cases} \tag{4.38}$$

When having drawn $\boldsymbol{z}^{(m+1)}$, step m of the MH sampling scheme is concluded. Similar to the Gibbs sampler, we discard the first N_{burn} burn-in simulations and decorrelate the Markov chain by only selecting every n_{thin}th observation. The number of stored simulations is taken as N_{sim}, which brings the total number of simulations to $N_{burn} + N_{sim} \cdot n_{thin}$. The posterior mean and variance are eventually taken as the mean and variance of the N_{sim} observations that we stored from the Markov chain. The complete MH sampling scheme is provided in Algorithm 3 in Appendix A.3.

4.4.7 Variational Logistic Regression

The third and last considered Bayesian method is Variational Logistic Regression (VLR). This method is very different from the Gibbs and MH samplers, since it is a deterministic approximation approach. VLR aims to maximize a lower bound on the marginal likelihood of the logit model. The explanation of the methodology is based on the work of Bishop (2006).

As a quick reminder, in a Bayesian framework we are aiming to yield posterior density $p(\boldsymbol{\beta}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{\beta})p(\boldsymbol{\beta})$. The variational approach proposes approximating conditional likelihood $p(\boldsymbol{y}|\boldsymbol{\beta})$ by a lower bound, such that we are able to approximate the posterior distribution as a Gaussian distribution. First note that the conditional distribution for y_i can be rewritten as:

$$\begin{aligned} p(y_i|\boldsymbol{\beta}) &= G(\boldsymbol{x}_i'\boldsymbol{\beta})^{y_i}(1 - G(\boldsymbol{x}_i'\boldsymbol{\beta}))^{1-y_i} = \left(\frac{e^{\boldsymbol{x}_i'\boldsymbol{\beta}}}{1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}}}\right)^{y_i}\left(\frac{1}{1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}}}\right)^{1-y_i} \\ &= e^{\boldsymbol{x}_i'\boldsymbol{\beta}\cdot y_i}\frac{1}{1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}}} = e^{\boldsymbol{x}_i'\boldsymbol{\beta}\cdot y_i}G(-\boldsymbol{x}_i'\boldsymbol{\beta}) \end{aligned} \tag{4.39}$$

Bishop (2006) has derived a variational lower bound for the logistic CDF $G(\boldsymbol{x}_i'\boldsymbol{\beta})$, by introducing variational parameter vector $\boldsymbol{\xi}$ of length N. This variational lower bound should be optimized

[4]We sample efficiently from the independent truncated logistic distribution by the inversion method described by Held & Holmes (2006).

with respect to $\boldsymbol{\xi}$ in order to yield the best approximation of the posterior distribution. For every observation i, λ is a function of ξ_i, that can be written as:

$$\lambda(\xi_i) = -\frac{1}{2\xi_i}\left(G(\xi_i) - \frac{1}{2}\right) \tag{4.40}$$

For all values of $\boldsymbol{\xi}$, the lower bound for $G(\boldsymbol{x}_i'\boldsymbol{\beta})$ is given by:

$$G(\boldsymbol{x}_i'\boldsymbol{\beta}) \geq G(\xi_i)e^{\frac{1}{2}(\boldsymbol{x}_i'\boldsymbol{\beta}-\xi_i)-\lambda(\xi_i)((\boldsymbol{x}_i'\boldsymbol{\beta})^2-\xi_i^2)} \tag{4.41}$$

This means, that we have found a lower bound for the conditional distribution for y_i:

$$\begin{aligned} p(y_i|\boldsymbol{\beta}) &= e^{\boldsymbol{x}_i'\boldsymbol{\beta}\cdot y_i}G(-\boldsymbol{x}_i'\boldsymbol{\beta}) \\ &\geq e^{\boldsymbol{x}_i'\boldsymbol{\beta}\cdot y_i}G(\xi_i)e^{-\frac{1}{2}(\boldsymbol{x}_i'\boldsymbol{\beta}+\xi_i)-\lambda(\xi_i)((\boldsymbol{x}_i'\boldsymbol{\beta})^2-\xi_i^2)} \\ &= h(\boldsymbol{x}_i, \xi_i) \end{aligned} \tag{4.42}$$

Therefore, a lower bound for $p(\boldsymbol{y}|\boldsymbol{\beta})p(\boldsymbol{\beta})$ can be formulated. In order to simplify the expression, we take the natural logarithm of this expression. The lower bound $\mathcal{L}(\boldsymbol{\xi})$ then becomes:

$$log\left(p(\boldsymbol{y}|\boldsymbol{\beta})p(\boldsymbol{\beta})\right) \geq log(p(\boldsymbol{\beta})) + \sum_{i=1}^{N} h(\boldsymbol{x}_i, \xi_i) = \mathcal{L}(\boldsymbol{\xi}) \tag{4.43}$$

When we would be able to write $\mathcal{L}(\boldsymbol{\xi})$ as a quadratic function of $\boldsymbol{\beta}$, we can obtain the corresponding approximation to the posterior distribution. Due to the posterior distribution of $\boldsymbol{\beta}$ being proportional to $p(\boldsymbol{y}|\boldsymbol{\beta})p(\boldsymbol{\beta})$, we can ignore all terms that are not a function of $\boldsymbol{\beta}$. Remember that we have defined the prior of the logit model as $P^L(\boldsymbol{\beta}) \sim N(\tilde{\boldsymbol{\beta}}^L, \tilde{\boldsymbol{B}}^L)$. By working out the expressions of Equation 4.43, we eventually get:

$$\begin{aligned} log\left(p(\boldsymbol{y}|\boldsymbol{\beta})p(\boldsymbol{\beta})\right) \geq &-\frac{1}{2}(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}}^L)'(\tilde{\boldsymbol{B}}^L)^{-1}(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}}^L) \\ &+\sum_{i=1}^{N}\left(\boldsymbol{x}_i'\boldsymbol{\beta}(y_i-\frac{1}{2}) - \lambda(\xi_i)\boldsymbol{\beta}'(\boldsymbol{x}_i\boldsymbol{x}_i')\boldsymbol{\beta}\right) + \text{constant} \end{aligned} \tag{4.44}$$

As this lower bound is a quadratic function of $\boldsymbol{\beta}$, we can identify a Gaussian variational posterior of the form $p(\boldsymbol{\beta}|\boldsymbol{y}) = N(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{B}})$, where the parameters are defined as:

$$\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{B}}\left((\tilde{\boldsymbol{B}}^L)^{-1}\tilde{\boldsymbol{\beta}}^L + \sum_{i=1}^{N}(y_i - \frac{1}{2})\boldsymbol{x}_i\right) \tag{4.45}$$

$$\hat{\boldsymbol{B}} = \left((\tilde{\boldsymbol{B}}^L)^{-1} + 2\sum_{i=1}^{N}\lambda(\xi_i)\boldsymbol{x}_i\boldsymbol{x}_i'\right)^{-1} \tag{4.46}$$

The only thing that is left for us, is to determine the optimal value of variational parameter vector $\boldsymbol{\xi}$ that maximizes lower bound $\mathcal{L}(\boldsymbol{\xi})$. This optimization process is slightly different for batch and sequential learning. We start by reviewing the methodology in a batch learning context. This is done by the Expectation-Maximization (EM) algorithm. As the name prescribes, this algorithm

consists of two steps: the Expectation step (E-step) and the Maximisation step (M-step). The algorithm constantly iterates between these two steps until convergence is reached.

The algorithm is initialized with $\hat{\boldsymbol{\beta}}^{old} = \tilde{\boldsymbol{\beta}}^{L}$ and $\hat{\boldsymbol{B}}^{old} = \tilde{\boldsymbol{B}}^{L}$. We then start with the M-step, where we solve the first order condition of $\mathcal{L}(\boldsymbol{\xi})$ with respect to $\boldsymbol{\xi}$. This allows us to express the locally optimal value of $\boldsymbol{\xi}$ in terms of $\hat{\boldsymbol{\beta}}^{old}$ and $\hat{\boldsymbol{B}}^{old}$ by:

$$\xi_i = \sqrt{\boldsymbol{x}_i'(\hat{\boldsymbol{B}} + \hat{\boldsymbol{\beta}}\hat{\boldsymbol{\beta}}')\boldsymbol{x}_i} \tag{4.47}$$

Then follows the E-step, where we calculate $\hat{\boldsymbol{\beta}}^{new}$ and $\hat{\boldsymbol{B}}^{new}$ using Equations 4.45 and 4.46. These values are then used for computing the new value of ξ_i in the M-step. We define that convergence in the EM algorithm has been reached when the maximum absolute difference between entries of $\hat{\boldsymbol{\beta}}^{new}$ and $\hat{\boldsymbol{\beta}}^{old}$ is smaller than 0.001.

The sequential approach to VLR is comparable to the batch approach. The formulas for posterior parameters $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{B}}$ remain the same. The only difference is that variational parameter ξ_i is not iteratively optimized using the EM algorithm, but is computed only once. Generally, the two approaches should approximately lead to the same results. However, different results might occur due to posterior distributions being approximated by a Gaussian distribution at every iteration of the sequential updating and ξ_i not being optimized using the EM algorithm.

In the sequential VLR approach, we initialize the distribution using the prior and as each data point is absorbed, lower bound $\mathcal{L}(\xi_i)$ is optimized using the optimal variational parameter defined in Equation 4.47. At every iteration, the posterior distribution is defined by $p(\boldsymbol{\beta}|\boldsymbol{y_i}) = N(\hat{\boldsymbol{\beta}}^{(i)}, \hat{\boldsymbol{B}}^{(i)})$. The posterior of the previous iteration is taken as the prior of the next iteration. At every iteration we define $\hat{\boldsymbol{\beta}}^{(i)}$ and $\hat{\boldsymbol{B}}^{(i)}$ by:

$$\hat{\boldsymbol{\beta}}^{(i)} = \hat{\boldsymbol{B}}^{(i)} \left(\left(\hat{\boldsymbol{B}}^{(i-1)} \right)^{-1} \hat{\boldsymbol{\beta}}^{(i-1)} + (y_i - \frac{1}{2})\boldsymbol{x}_i \right) \tag{4.48}$$

$$\hat{\boldsymbol{B}}^{(i)} = \left(\left(\hat{\boldsymbol{B}}^{(i-1)} \right)^{-1} + 2 \sum_{i=1}^{N} \lambda(\xi_i) \boldsymbol{x}_i \boldsymbol{x}_i' \right)^{-1} \tag{4.49}$$

This leads to the schemes for batch and sequential VLR, which are provided in Algorithms 4 and 5 in Appendix A.4.

4.4.8 Validation and model performance

For the validation of Bayesian BAIT, we follow the manipulated K-fold cross-validation (CV) approach, as proposed by Schrama (2021). The technology is validated on the real-life data set. General K-fold cross-validation implies randomly partitioning the dataset in K parts. The validation process then consists of K iterations, where at every iteration the model is trained on $K-1$ parts of the data set and tested on the one remaining part. The mean of the performances on these K respective test sets is reported as the final performance. In the manipulated K-fold CV, we manipulate the K-fold CV such that at every iteration the selection of the holdout sample is controlled. This approach is necessary since the real-life data set consists of various decisions made for the same scenario. This means that multiple observations have identical $\boldsymbol{x}_i$. Because we want the test set to weigh the performance on every scenario equally, every scenario should be included

only once.

Model performance is measured based on how well the predicted model probabilities relate to the decisions made in reality. We define $\hat{f}_i$ as the predicted probability for decision task i. For the logit and probit models, these predicted probabilities are respectively given by $\hat{f}_i^L = G(\boldsymbol{x}_i'\hat{\boldsymbol{\beta}})$ and $\hat{f}_i^P = \Phi(\boldsymbol{x}_i'\hat{\boldsymbol{\beta}})$.

To compare the predictive performance of the various Bayesian BAIT approaches, we use four metrics: accuracy (ACC), precision (PREC), recall (REC) and Matthews correlation coefficient (MCC). To measure the calibration performance of the approaches, we consider the calibration error (CE). We use the description provided by Saito & Rehmsmeier (2015) to explain the predictive performance metrics. Accuracy, recall, precision and MCC are all widely used in binary classification problems. The real decisions made in the test set can be divided into two different classes, positives and negatives. BAIT classifies the samples in the test set as either positive or negative, given a certain threshold T. We set this threshold default to 0.5. We define the predicted decision of BAIT by:

$$\hat{y}_i = \begin{cases} 1 & \text{if } \hat{f}_i > T \\ 0 & \text{if } \hat{f}_i \leq T \end{cases} \tag{4.50}$$

To evaluate the performance, the actual decision made by an expert is compared to the predicted decision that BAIT would make. This comparison is made using a confusion matrix, which separates the decisions into four types of outcomes: two types of correct predictions, true positives (TP) and true negatives (TN), and two types of incorrect predictions, false positives (FP) and false negatives (FN). Accuracy, precision and recall are three straightforward performance metrics. Accuracy, also known as the hit rate, is defined as the fraction of correctly predicted decisions out of all decision tasks. Precision compares the fraction of true positives to false positives and recall compares true positives to false negatives. The definitions thus become:

$$\text{ACC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{4.51}$$

$$\text{PREC} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4.52}$$

$$\text{REC} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4.53}$$

The advantages of these metrics are that they are widely known and straightforward to interpret. Nevertheless, these metrics only generate reliable results for balanced data sets. They might lead to misleading conclusions for imbalanced data sets, as they do not consider the ratio between positive and negative elements (Chicco & Jurman, 2020). The Matthews correlation coefficient (MCC) is an approach that pays more respect to this ratio. One can only obtain a high MCC when the classifier makes correct predictions on both the majority of the negative cases and the majority of the positive cases. The coefficient gives an indication of how much better a given prediction is than a random one. MCC is defined on the interval $[-1, 1]$. MCC = 1 indicates a perfect agreement between prediction and observation, MCC = 0 is expected for a prediction no better than random, and MCC = -1 indicates total disagreement between prediction and observation (Matthews, 1975).

The formula is provided by:

$$\text{MCC} = \frac{\text{TP} \cdot \text{TN} - \text{FP} \cdot \text{FN}}{\sqrt{(\text{TP} + \text{FP}) \cdot (\text{TP} + \text{FN}) \cdot (\text{TN} + \text{FP}) \cdot (\text{TN} + \text{FN})}} \tag{4.54}$$

The last performance metric we discuss is the calibration performance metric. A shortcoming of the predictive performance metrics is that these metrics are based on the confusion matrix, which assumes a threshold. If we assume a threshold of 0.5, a predicted probability of 0.49 would be classified as 0 and a predicted probability of 0.51 would be classified as 1. However, in the context of BAIT, a model output of 0.51 should be interpreted as "51% of the experts would make a positive decision". In other words, we desire that the predicted decision probabilities are reflective of the true underlying probability of the population. That is, the predicted class probability needs to be well calibrated (Kuhn & Johnson, 2013). For this reason, we also analyze the calibration errors (CE) of the approaches.

Caruana & Niculescu-Mizil (2004) have found that the correlation of CE with other performance metrics is relatively small. One can achieve excellent performance on predictive performance metrics and be extremely poor calibrated. Conversely, models might achieve good calibration, yet have poor predictive performance. Therefore, we believe the performance of BAIT should not be reviewed on one particular metric, but on a combination of calibration and the predictive performance metrics.

A description of how CE should be calculated is based on the work of Caruana & Niculescu-Mizil (2004). Let us denote y as the vector of actual made decisions and $\hat{f}$ is the vector of predicted probabilities by BAIT. N_{test} and N_{bin} are defined as the sizes of the test set and the bin size respectively and give us the number of bins $\#bins = N_{test} - N_{bin} + 1$. First, we order all scenarios by their predicted value and make bins of scenarios $\{1, \ldots, N_{bin}\}, \{2, \ldots, N_{bin} + 1\}, \ldots, \{\#bins, \ldots, N_{test}\}$. For every bin we calculate the percentage of the cases that were positive in the real-life data set by $\overline{y}_b = \frac{1}{N_{bin}} \sum_{i=b}^{b+N_{bin}-1} y_i$. This observed frequency should approximate the true probability that these cases are positive. We compare this to the mean predicted probability for these cases, given by $\overline{f}_b = \frac{1}{N_{bin}} \sum_{i=b}^{b+N_{bin}-1} \hat{f}_i$. The absolute difference between the observed frequency and the mean predicted probability is called the calibration error. The smaller the calibration error, the better a prediction is calibrated around the true probability. The calibration error is thus defined as:

$$CE = \frac{1}{\#bins} \sum_{b=1}^{\#bins} |\overline{y}_b - \overline{f}_b| \tag{4.55}$$

4.5 Results for the special welfare applications

This section focuses on the results of the special welfare applications data set. First, we discuss the results for the prior parameters, based on the expert beliefs. After this follows a review of the conditions under which we applied the Bayesian methods. Next, we evaluate the resulting posterior distributions for all of the models. The section is concluded by a comparison of the model performances.

4.5.1 Prior parameters

To obtain the hyperparameters of the prior distribution, we require input parameters of experts. The required input parameters are P_{base} (the base probability that $y = 1$), the expected marginal effects (EMEs) of variables and the confidence intervals of these inputs. Nevertheless, we have not requested these inputs directly from the experts. This has two reasons. Firstly, BAIT should be an accessible technology and thus we do not want to burden experts by requesting too much complex information. Secondly, experts have not been educated to make such numeric assessments, which would make assessments both subjective and doubtful. As an alternative, we have developed our own methodology for obtaining the input parameters.

In the first place, we have chosen P_{base} to be 0.5. This implies that a priori we expect $P[y = 1]$ and $P[y = 0]$ to be equally large. We take this impartial position since there is no prior information to make any extreme assumptions on the base probability. The default for the 95% confidence interval of P_{base} has been set to $[0.3, 0.7]$. We consider this as a rather wide interval, which corresponds to the uncertainty we have with respect to the estimated base probability.

Secondly, for the inputs of the EMEs, we use the expected importance of variables as stated in Table 3.6. For the three descending levels "critical", "important" or "nice to have", we set the EME to 15%, 10% and 5% respectively. The default 95% confidence interval of these EMEs is set to $[\text{EME} - 10\%, \text{EME} + 10\%]$. Taking a fixed percentage as uncertainty on these EMEs ensures that every parameter holds a comparable prior uncertainty. We aim for these intervals to be rather wide, since we do not want a too informative prior.

Using these input values, we obtain the hyperparameters stated in Table 4.2[5]. The table shows that the linear trend in the EMEs is transferred to a linear trend in the location parameters, as we desired. The variance parameters are relatively constant over the EMEs, which is in line with our expectations based on the confidence interval of the EME.

Taking into account the sign and importance of the variables in the special welfare data set, we obtain the prior parameters that are displayed in Table 4.3. Here it is important to note how we have dealt with the expected marginal effect of ordinal variables with non-linear levels, i.e. *Var 1* and *Var 2*. These variables are dummy encoded, which means that we have to take one variable as a base category to avoid perfect multicollinearity. As it is customary to take the most frequently appearing category as base category, these are *level 1* and *level 0* for *Var 1* and *Var 2* respectively. The effect of the dummy variables should be set with respect to the base category. As *Var 1* is believed to have a positive effect on the dependent variable, *Var 1 - level 0* is believed to have a negative impact with respect to the base category, whereas *Var 1 - level 2* has a positive impact. The effect of *Var 2 - level 1* and *Var 2 - level 2* remain positive with respect to the base category.

Furthermore, the stated importance level corresponds to the situation where the ordinal variable would move from one extreme on the ordinal scale to the other. However, if we move from one extreme to the intermediate value, the importance level is set to one level lower. For example, *Var 1* is labelled as a critical variable. When we move from the base category to *level 0* or *level 2*, this is considered an intermediate step. Therefore, their importance is labelled as one importance level lower, i.e. important. As *Var 2* is already assigned the lowest importance, the effect of *level 1* and *level 2* are both assigned the lowest importance.

[5]To verify the correct computation of our prior parameters, we check the approximate scaling factor of $\beta^L \approx 1.6\beta^P$ as proposed by Amemiya (1981). Our prior parameters are in line with this.

Table 4.2: Hyperparameters of prior distributions based on EMEs

Importance	Expected marginal effect	$\tilde{\beta}^L_{cont}$	$\tilde{\sigma}^L_{cont}$	$\tilde{\beta}^L_{dum}$	$\tilde{\sigma}^L_{dum}$	$\tilde{\beta}^P_{cont}$	$\tilde{\sigma}^P_{cont}$	$\tilde{\beta}^P_{dum}$	$\tilde{\sigma}^P_{dum}$
Critical	0.15	0.600	0.204	0.619	0.214	0.376	0.128	0.385	0.133
Important	0.10	0.400	0.204	0.405	0.207	0.251	0.128	0.253	0.129
Nice to have	0.05	0.200	0.204	0.201	0.204	0.125	0.128	0.126	0.128

Table 4.3: Parameters of prior distributions for the special welfare data set

Variable	Importance	Sign	$\tilde{\beta}^L$	$\tilde{\sigma}^L$	$\tilde{\beta}^P$	$\tilde{\sigma}^P$
Constant			-0.846	0.432	-0.528	0.268
Var 1 - level 0	Important	Negative	-0.405	0.207	-0.253	0.129
Var 1 - level 2	Important	Positive	0.405	0.207	0.253	0.129
Var 2 - level 1	Nice to have	Positive	0.201	0.204	0.126	0.128
Var 2 - level 2	Nice to have	Positive	0.201	0.204	0.126	0.128
Var 3	Important	Positive	0.405	0.207	0.253	0.129
Var 4	Critical	Positive	0.619	0.213	0.385	0.133
Var 5	Important	Positive	0.400	0.204	0.251	0.128

4.5.2 Evaluation of prior inputs

In this section, we briefly evaluate our choice for the prior inputs. This evaluation is carried out by computing the base probability and marginal effects (MEs) of variables on the experimental data set. The base probability is taken as the mean of all experimental decisions, which is equal to 0.204. This is substantially different from the impartial base probability of 0.5 which we assumed.

Based on this realised base probability, we computed the MEs of the variables using the parameters of Maximum Likelihood Estimation (MLE) on the experimental data set. These MEs are compared to the expected marginal effect (EME) we assumed. Moreover, we also compare the expected confidence intervals, with lower bounds LBEME and upper bounds UBEME, to the realised confidence interval, which is denoted by lower bound LBME and upper bound UBME. Table 4.4 provides the prior EMEs and the realised MEs for the logit and probit model with their respective 95% confidence interval.

From the table, we yield several findings. For instance, we can see that the marginal effects of the logit and probit model are very aligned, as desired. In retrospect, the EME of critical variable *Var 4* was chosen too conservative, since the realised MEs are a factor four times larger. The realised MEs of the important variables are (in absolute terms) also larger than we expected. For the "nice to have" variables, no straightforward conclusion on the magnitude of the EMEs can be drawn. Moreover, the confidence intervals of the realised MEs are not as balanced as those of the

Table 4.4: True marginal effects and confidence intervals for the special welfare data set

		Expected			Logit			Probit		
	Importance	EME	LBEME	UBEME	ME	LBME	UBME	ME	LBME	UBME
Var 1 - level 0	Important	-0.10	-0.20	0	-0.193	-0.202	-0.156	-0.2	-0.204	-0.168
Var 1 - level 2	Important	0.10	0	0.20	0.124	-0.059	0.379	0.111	-0.057	0.331
Var 2 - level 1	Nice to have	0.05	-0.05	0.15	-0.006	-0.117	0.186	-0.002	-0.118	0.177
Var 2 - level 2	Nice to have	0.05	-0.05	0.15	0.095	-0.066	0.329	0.09	-0.066	0.297
Var 3	Important	0.10	0	0.20	0.567	0.298	0.714	0.536	0.3	0.695
Var 4	Critical	0.15	0.05	0.25	0.639	0.339	0.757	0.62	0.342	0.756
Var 5	Important	0.10	0	0.20	0.443	0.26	0.626	0.452	0.277	0.627

EMEs, which is caused by the magnitude of the base probability. In general, our choice of confidence intervals for the EME was quite narrow compared to the realised ME. From these results, we can conclude that our prior inputs were chosen rather conservative and informative.

4.5.3 Bayesian modelling conditions

Now we review the conditions under which we have executed the MCMC methods. We have set the number of burn-in simulation $N_{burn} = 1000$ and thinning value $n_{thin} = 5$. The total number of stored simulations N_{sim} is taken as 5000. The correctness of the implementation of Bayesian software can be verified using the methodology of Cook et al. (2006). A confirmation that we have correctly implemented the Bayesian software for the Gibbs and MH algorithms can be found in Appendix B. Furthermore, Appendix C.1 provides plots of the Markov chains for all models. The plots show that the Markov chains from which we sample values of β have all converged.

For the MH algorithm, we should account for enough candidate values being accepted. Therefore, the mean acceptance ratios should be as close to 1 as possible. For the two sequential samplers, we have taken the mean of the acceptance ratios for every step of the sequential updating. We have obtained acceptance ratios of 0.949, 0.661 and 0.983 for respectively the *MLE-based sequential Bayesian* approach, *batch Bayesian* approach and *batch - sequential* Bayesian approach. The acceptance ratios for the sequential approaches are close to 1, because the posterior is only updated on one data point per iteration. This means that the candidate $\boldsymbol{\beta}$'s are close to the old $\boldsymbol{\beta}$'s. For the batch approach, this is not the case, which explains the lower acceptance ratio.

Moreover, a correct implementation of the batch VLR algorithm should be accompanied by a low number of iterations until convergence of the EM algorithm (Bishop, 2006). Since the EM algorithms of the *batch Bayesian* and *batch - sequential Bayesian* approaches respectively required 5 and 6 iterations until convergence, there are no signs of incorrect implementation.

4.5.4 Posterior distributions

In this section we provide a comparative analysis of the posterior distributions for all models. Table 4.5 shows an overview of all posterior parameters. Figure 4.1 displays the plots of posterior distributions of *Var 5* for each of these models. In this figure, we have randomly chosen to display *Var 5*. Appendix C.2 provides plots of the posterior distributions of other variables. We make five key observations based on these results.

The first observation from these plots is that posterior distributions among different Bayesian methods behave in roughly the same way for the same model settings. The posteriors of the fully Bayesian approaches lie very close to each other. These posteriors satisfy the desired averaging of the prior expert beliefs and the data, but are slightly more impacted by the prior[6]. The posterior distributions of *MLE sequential Bayes* are close to the MLE distribution. However, the posterior distributions of the MCMC methods remain closer to the MLE distribution than those of VLR.

Secondly, no large approximation errors arise due to the sequential updating of the posteriors in the *batch - sequential Bayesian* approaches. This is verified by comparing to the benchmark

[6]We have verified the correct implementation of our Bayesian approaches once again by a stepwise evaluation of the posteriors. We have seen that the posteriors start at the expert prior belief and as data points are absorbed, the density moves more towards the MLE. Moreover, by replicating the data set a large number of times, the prior inputs become neglected and the posterior distributions become asymptotically equivalent to the MLE.

of the *batch Bayesian* approaches. The minor differences between these distributions arise due to approximating the posterior by a Gaussian distribution at every iteration of the sequential updating.

Thirdly, the fully Bayesian posterior densities have relatively small absolute means and low standard deviations. This is caused by the informative and conservative choice of our prior distribution. That the prior is informative can be derived from the curve of the prior distributions being less wide than those of the MLE. This had led to small standard deviations of the fully Bayesian posteriors, relative to MLE-based Bayesian posteriors. Moreover, the absolute prior means were chosen to be rather small relative to the MLE estimate. Due to the informativeness of the prior and the limited amount of data, fully Bayesian posterior means stayed close to the prior means and remained small in absolute terms.

Another result of our choice distribution is that a priori *important* or *critical* variables remain significant for the fully Bayesian approaches. This can be seen in *Var 1 - level 2*. This variable is insignificant in the MLE-based approaches. However, the prior label as *important* translates into the variable being significant in our prior density. Because of the informative prior and limited amount of data, the effect of the data is not large enough to push the posterior distribution of these variables to become insignificant.

Lastly, the scaling of parameters between the three Bayesian methods is as we would expect. Logit MH and VLR are both based on the logit model and lead to similar mean and variance estimates. There is a scaling difference between the parameters of the probit and logit model, where the scaling factor ranges between 1 and 2.

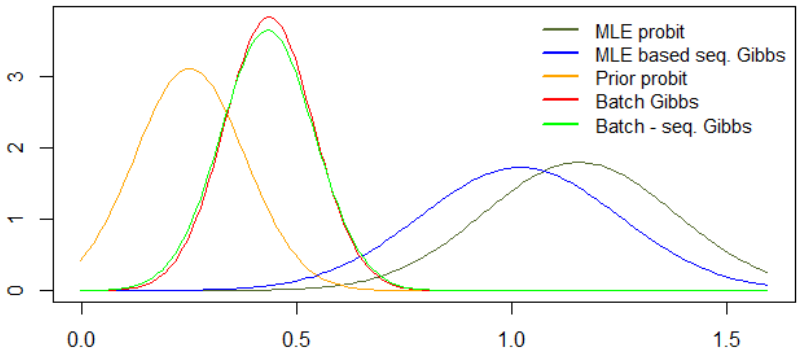
Table 4.5: Parameters of posterior distributions in the special welfare data set

	Probit MLE		MLE-based sequential Gibbs		Batch Gibbs		Batch - sequential Gibbs	
	$\hat{\beta}$	$\hat{\sigma}$	$\hat{\beta}$	$\hat{\sigma}$	$\hat{\beta}$	$\hat{\sigma}$	$\hat{\beta}$	$\hat{\sigma}$
Constant	-2.043***	0.278	-1.747***	0.325	-1.301***	0.132	-1.276***	0.137
Var 1 - level 0	-1.006***	0.301	-0.945**	0.405	-0.406***	0.108	-0.412***	0.112
Var 1 - level 2	0.052	0.232	-0.093	0.269	0.242**	0.101	0.239**	0.106
Var 2 - level 1	-0.170	0.236	-0.168	0.255	-0.031	0.106	-0.043	0.109
Var 2 - level 2	-0.074	0.203	-0.194	0.217	0.073	0.103	0.074	0.106
Var 3	0.877***	0.186	0.754***	0.19	0.409***	0.1	0.399***	0.107
Var 4	0.965***	0.221	0.870***	0.248	0.429***	0.103	0.422***	0.107
Var 5	1.158***	0.222	1.018***	0.231	0.436***	0.106	0.428***	0.109
	Logit MLE		MLE-based sequential MH		Batch MH		Batch - sequential MH	
	$\hat{\beta}$	$\hat{\sigma}$	$\hat{\beta}$	$\hat{\sigma}$	$\hat{\beta}$	$\hat{\sigma}$	$\hat{\beta}$	$\hat{\sigma}$
Constant	-3.575***	0.517	-2.862***	0.596	-2.093***	0.219	-2.141***	0.24
Var 1 - level 0	-1.938***	0.555	-1.603***	0.602	-0.656***	0.173	-0.660***	0.177
Var 1 - level 2	0.001	0.423	-0.122	0.430	0.389**	0.161	0.429***	0.171
Var 2 - level 1	-0.190	0.422	-0.348	0.405	-0.048	0.176	-0.034	0.153
Var 2 - level 2	-0.108	0.366	-0.216	0.443	0.124	0.165	0.132	0.152
Var 3	1.506***	0.341	1.243***	0.401	0.655***	0.161	0.685***	0.187
Var 4	1.769***	0.414	1.409***	0.425	0.696***	0.168	0.706***	0.155
Var 5	2.092***	0.413	1.547***	0.455	0.694***	0.168	0.704***	0.186
	MLE-based sequential VLR		Batch VLR		Batch - sequential VLR			
	$\hat{\beta}$	$\hat{\sigma}$	$\hat{\beta}$	$\hat{\sigma}$	$\hat{\beta}$	$\hat{\sigma}$		
Constant	-3.531***	0.567	-2.037***	0.203	-2.045***	0.203		
Var 1 - level 0	-1.870***	0.597	-0.624***	0.170	-0.631***	0.171		
Var 1 - level 2	0.073	0.432	0.349**	0.162	0.352**	0.162		
Var 2 - level 1	-0.193	0.438	-0.025	0.167	-0.028	0.167		
Var 2 - level 2	-0.173	0.396	0.125	0.162	0.128	0.163		
Var 3	1.496***	0.382	0.595***	0.158	0.596***	0.158		
Var 4	1.751***	0.442	0.627***	0.163	0.622***	0.163		
Var 5	2.075***	0.425	0.642***	0.167	0.642***	0.167		

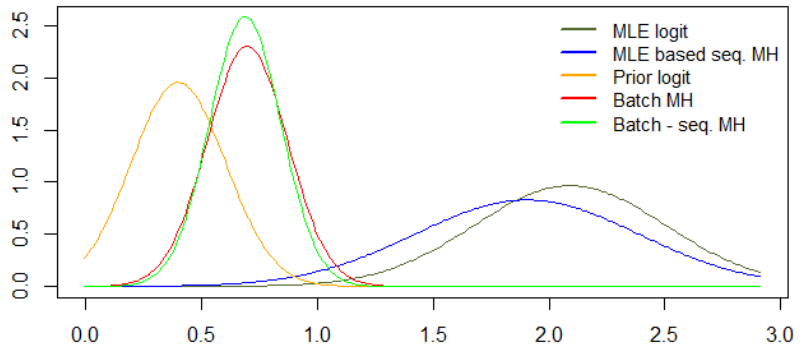
Notes: ***, **, * indicate statistical significance at the 0.01, 0.05, 0.10 levels, respectively.

Figure 4.1: The posterior distributions of the effect of *Var 5* in the probit Gibbs, logit MH and VLR models for the special welfare data set

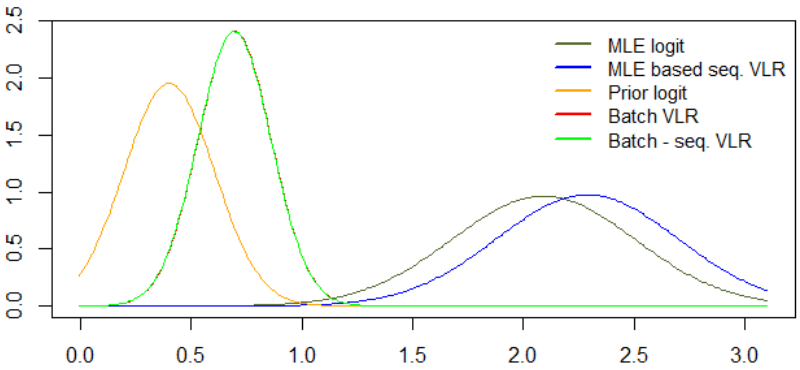
(a) Probit MLE and posterior distributions of probit Gibbs samplers



(b) Logit MLE and posterior distributions of logit MH samplers



(c) Logit MLE and posterior distributions of VLR



4.5.5 Model performance

Table 4.9 provides the performance matrix, based on a manipulated K-fold cross-validation with $K = 10$. For every fold, the real-life data set of 67 observations was randomly split into a training set of 44 observations and a test set of 23. This implies an approximate $\frac{2}{3}/\frac{1}{3}$ training-test split for every fold. For the predictive performance metrics, we set the classification threshold to default value 0.5. Notice that in the table, we provide the averaged performance over the 10 folds. In brackets behind the average performances, we indicate for how many folds that specific model achieved the best performance on that specific metric. Because many models predicted exactly the same decisions,

there were many shared winners for the predictive performance metrics.

An interesting first observation is that we can make three groups of models with comparable performances. The first group consists of the MLE approaches, which perform exactly the same. Secondly, we have the group of fully Bayesian approaches, consisting of *batch Bayesian* and *batch - sequential Bayesian* models. These provide almost identical performances due to their similar posterior distributions. Lastly, the group of *MLE-based sequential Bayesian* approaches is the only group that has slightly different performances for different Bayesian methods.

From here, we discuss each of the predictive performance metrics briefly. The first performance metric we focus on is accuracy, i.e. the fraction of correct classifications. Overall, all models have a very comparable accuracy, ranging between 0.743 and 0.752. Only the *logit MLE-based sequential MH* model noted a slightly lower accuracy. The models with the highest average accuracy and highest number of winning folds on this data set are the MLE and fully Bayesian approaches.

Secondly, the recall metric tells us the ratio of true positive predictions relative to all positive decisions. Recall shows slightly more variation than accuracy. For this data set, the *MLE-based sequential VLR* model provides the best recall and most wins. The two MLE models also provide a good recall metric. The fully Bayesian approaches have the worst performance on recall.

Precision is defined as the ratio of true positive predictions relative to the all positive predictions. Interestingly, a decreasing recall is accompanied by an increasing precision. On this data set, the fully Bayesian approaches have the highest precision and highest number of wins, whereas *MLE-based sequential VLR* has the lowest precision and lowest number of wins.

The last predictive performance metric we review is Matthews Correlation Coefficient (MCC). This metric does not deviate a lot among the various approaches, but tells us that all models perform better than a random prediction. The best performing models on MCC for this data set are the fully Bayesian models. The group of *MLE sequential Bayes* models perform the worst on MCC.

Altogether, the predictive performance metrics do not provide one general conclusion. Therefore, we explore whether the calibration error (CE) provides more conclusive results. CE tells us how closely the predicted probabilities reflect the true probabilities of experts. For computing the CE, we should determine the number of bins. There should be enough observations in one bin, but also enough bins to average over. Since we only have 23 test observations, we decided on a bin size of 8 observations. This leaves us with 16 bins per fold. In Table 4.9, we report the average CE over all bins and folds for every model. Whereas the two MLE approaches have the best (i.e. lowest) average CE for this data set, the fully Bayesian methods have the worst average CE.

To get a better intuition on what CEs entail, Figure 4.2 provides calibration plots of the best calibrated model, MLE logit, and the worst calibrated model, *Batch - sequential MH*. An overview of all calibration plots can be found in Appendix C.3. In the calibration plots, the grey dots show how the mean predicted probabilities per bin compare to the mean true probabilities per bin. The black line indicates the ideal situation where the two quantities are equivalent. CE is computed as the mean absolute deviation of a point with respect to this line. The good CE value of the MLE logit model can be observed from the grey dots following the ideal line closely. This is due to the model's high posterior parameter values. The bad CE of *Batch - sequential MH* can be explained by the predicted probabilities being centred around the middle and large CEs in the extremes. This is due to the low absolute posterior means, which are caused by the informative prior distributions with low absolute means.

The last performance metric is the average computation time per fold. MLE and VLR are very fast methods, since these do not need any simulation of variables. Probit Gibbs and logit MH take considerably longer, as they require the simulation of a large number of parameters. The batch approaches require simulating only one MCMC chain, which causes the computation time to remain within reasonable bounds. Nevertheless, in the sequential Bayesian approaches, the simulation of a Markov chain at every iteration adds up to a considerable computation time.

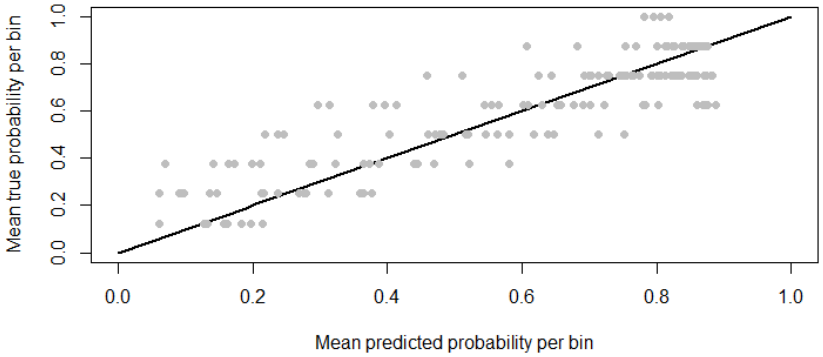
Table 4.6: Performances per model for the special welfare data set

Name model	Accuracy	Recall	Precision	MCC	Cal. error	Times
MLE probit	0.752 (6)	0.822 (8)	0.759 (2)	0.493 (5)	0.101 (0)	0:01
MLE-based sequential Gibbs	0.743 (4)	0.802 (5)	0.761 (2)	0.478 (3)	0.116 (2)	7:51
Batch Gibbs	0.752 (6)	0.727 (1)	0.823 (9)	0.526 (6)	0.186 (0)	0:14
Batch - sequential Gibbs	0.748 (5)	0.719 (1)	0.821 (8)	0.518 (5)	0.189 (0)	8:00
MLE logit	0.752 (6)	0.822 (8)	0.759 (2)	0.493 (5)	0.097 (2)	0:01
MLE-based sequential MH	0.726 (3)	0.773 (4)	0.754 (2)	0.448 (3)	0.125 (2)	24:05
Batch MH	0.752 (6)	0.727 (1)	0.823 (9)	0.526 (6)	0.189 (0)	0:38
Batch - sequential MH	0.752 (6)	0.727 (1)	0.823 (9)	0.526 (6)	0.190 (0)	24:40
MLE-based sequential VLR	0.743 (5)	0.838 (10)	0.740 (1)	0.473 (4)	0.108 (4)	0:01
Batch VLR	0.752 (6)	0.727 (1)	0.823 (9)	0.526 (6)	0.188 (0)	0:01
Batch - sequential VLR	0.752 (6)	0.727 (1)	0.823 (9)	0.526 (6)	0.189 (0)	0:01

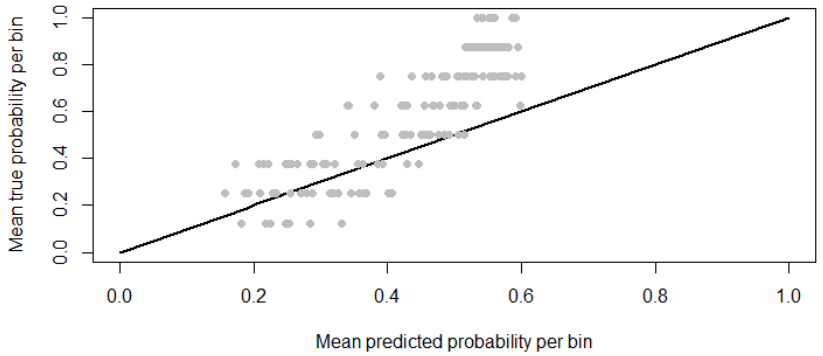
Notes: The numbers in between brackets indicate for how many folds this model resulted in the optimal performance.

Figure 4.2: Calibration plots of the best and worst calibrated models for the special welfare data set

(a) Calibration plot of MLE logit



(b) Calibration plot of Batch - sequential MH



4.6 Results for COVID-19 ICU uptakes

4.6.1 Prior parameters

The prior parameters for the COVID-19 ICU uptakes were set identical to the special welfare applications case for each type of independent variable. Hence, we have ended up with the same prior parameters as provided in Table 4.2. Taking into account the sign and importance as stated by the experts, we obtain the parameters of the prior distributions. The results are shown in Table 4.7. Here it is important to note how we have dealt with the expected marginal effect of variable *BMI*, the only ordinal variables with non-linear levels. This variable was encoded as base category, with BMI between 20 and 40 as base category. Since the variable is believed to be *n-shaped*, both the lower and higher category are expected to have a negative impact on the decision.

Table 4.7: Parameters of prior distributions for the COVID-19 data set

Variable	Importance	Sign	$\tilde{\beta}^L$	$\tilde{\sigma}^L$	$\tilde{\beta}^P$	$\tilde{\sigma}^P$
constant			0.992	0.432	0.626	0.268
Relocation possbilities	Nice to have	Positive	0.201	0.204	0.126	0.128
ICU capacity	Critical	Positive	0.619	0.214	0.385	0.133
Acute respiratory threat	Critical	Positive	0.619	0.204	0.385	0.133
Age	Critical	Negative	-0.600	0.204	-0.376	0.128
Comorbidity: Cognitive functioning	Critical	Negative	-0.600	0.204	-0.376	0.128
Comorbidity: Cardiovascular	Critical	Negative	-0.600	0.204	-0.376	0.128
Comorbidity: Pulmonary	Critical	Negative	-0.600	0.204	-0.376	0.128
Comorbidity: Renal	Critical	Negative	-0.600	0.204	-0.376	0.128
Comorbidity: Liver	Critical	Negative	-0.600	0.204	-0.376	0.128
Comorbidity: Immunity	Important	Negative	-0.400	0.204	-0.251	0.128
COVID pneumonia disease course	Nice to have	Negative	-0.201	0.204	-0.126	0.128
BMI <20	Nice to have	Negative	-0.201	0.204	-0.126	0.128
BMI 20 - 40	(base)	(base)	(base)	(base)	(base)	(base)
BMI >40	Nice to have	Negative	-0.201	0.204	-0.126	0.128
Frailty	Critical	Positive	0.619	0.214	0.385	0.133
Patient decision	Important	Positive	0.405	0.207	0.253	0.129

4.6.2 Posterior distributions

In the COVID-19 case, the same 11 models are considered as previously for the special welfare application case. The MCMC methods are computed under the same conditions, i.e. $N_{burn} = 1000$, $n_{thin} = 5$ and $N_{sim} = 5000$. Figure 4.8 provides the parameters of the posterior distribution for the COVID-19 data set.

We review whether the same conclusions can be drawn as for the special welfare applications case. Similar to the previous case, the MLE-based sequential Bayesian approaches are very much in line with the output of the MLE models. Secondly, the batch Bayesian models lead to very similar parameters estimates as the batch - sequential Bayesian models. Hence, no large approximation errors arise due to sequential updating of the parameters. Thirdly, we again see that our prior distribution was chosen rather informatively. Compared to the MLE, our self-designed priors have a smaller standard deviation. As an effect, the posterior standard deviations for the two fully Bayesian methods are smaller than for the MLE-based Bayesian method. Fourthly, a priori *critical* and *important* variables always remain significant in the fully Bayesian approaches. This can be

seen in variables such as *Comorbidity: Pulmonary* and *Comorbidity: Renal*. These variables are insignificant in the MLE-based approaches, but remain significant in the fully Bayesian approaches due to their prior label as *critical*. Lastly, the scaling factors between the parameters of the posterior distributions for the different Bayesian methods are as we would expect.

Table 4.8: Parameters of the posterior distribution for the COVID-19 data set

	Probit MLE		MLE-based sequential Gibbs		Batch Gibbs		Batch Gibbs	
	$\hat{\beta}$	$\hat{\sigma}$	$\hat{\beta}$	$\hat{\sigma}$	$\hat{\beta}$	$\hat{\sigma}$	$\hat{\beta}$	$\hat{\sigma}$
constant	0.832**	0.344	0.736**	0.345	0.960***	0.154	0.961***	0.156
Relocation possbilities	-0.374**	0.161	-0.321*	0.178	-0.081	0.091	-0.126	0.097
ICU capacity	0.524***	0.18	0.556***	0.214	0.375***	0.100	0.374***	0.102
Acute respiratory threat	-0.387**	0.181	-0.394*	0.206	0.088	0.102	0.091	0.105
Age	-0.796***	0.191	-0.753***	0.224	-0.488***	0.099	-0.459***	0.102
Comorbidity: Cognitive functioning	-0.360**	0.175	-0.340*	0.195	-0.328***	0.098	-0.338***	0.100
Comorbidity: Cardiovascular	-0.336*	0.190	-0.328*	0.185	-0.369***	0.098	-0.406***	0.100
Comorbidity: Pulmonary	-0.073	0.186	-0.064	0.164	-0.225**	0.099	-0.211**	0.101
Comorbidity: Renal	0.162	0.193	0.177	0.215	-0.221**	0.105	-0.199*	0.103
Comorbidity: Liver	-0.055	0.185	0.045	0.179	-0.192**	0.098	-0.214**	0.102
Comorbidity: Immunity	-0.158	0.226	-0.193	0.243	-0.28***	0.104	-0.273***	0.104
COVID pneumonia disease course	-0.249*	0.146	-0.278*	0.151	-0.209**	0.092	-0.199**	0.094
BMI <20	0.122	0.170	0.132	0.160	-0.037	0.094	0.015	0.098
BMI >40	0.099	0.186	0.173	0.248	-0.039	0.097	-0.025	0.098
Frailty	1.139***	0.239	1.110***	0.238	0.620***	0.105	0.559***	0.108
Patient decision	0.336**	0.150	0.320**	0.158	0.271***	0.092	0.292***	0.095
	Logit MLE		MLE-based sequential MH		Batch MH		Batch - sequential MH	
	$\hat{\beta}$	$\hat{\sigma}$	$\hat{\beta}$	$\hat{\sigma}$	$\hat{\beta}$	$\hat{\sigma}$	$\hat{\beta}$	$\hat{\sigma}$
constant	1.328**	0.582	1.398**	0.586	1.553***	0.250	1.562***	0.273
Relocation possbilities	-0.649**	0.285	-0.666**	0.292	-0.102	0.150	-0.159	0.172
ICU capacity	0.863***	0.308	0.918**	0.365	0.607***	0.164	0.590***	0.163
Acute respiratory threat	-0.658**	0.315	-0.661**	0.321	0.171	0.166	0.179	0.149
Age	-1.309***	0.338	-1.382***	0.368	-0.772***	0.164	-0.741***	0.153
Comorbidity: Cognitive functioning	-0.623**	0.301	-0.586**	0.272	-0.540***	0.162	-0.550***	0.147
Comorbidity: Cardiovascular	-0.541*	0.327	-0.555*	0.329	-0.610***	0.161	-0.613***	0.150
Comorbidity: Pulmonary	-0.111	0.311	-0.194	0.261	-0.376**	0.162	-0.324*	0.171
Comorbidity: Renal	0.296	0.331	0.276	0.346	-0.373**	0.164	-0.399**	0.161
Comorbidity: Liver	-0.11	0.316	-0.139	0.300	-0.326**	0.163	-0.319*	0.168
Comorbidity: Immunity	-0.197	0.381	-0.148	0.315	-0.446***	0.168	-0.478***	0.169
COVID pneumonia disease course	-0.411	0.252	-0.399	0.286	-0.326**	0.151	-0.324**	0.149
BMI <20	0.185	0.294	0.159	0.301	-0.079	0.159	-0.072	0.139
BMI >40	0.162	0.329	0.155	0.338	-0.081	0.160	-0.096	0.164
Frailty	1.962***	0.418	2.007***	0.386	1.009***	0.169	1.012***	0.160
Patient decision	0.607**	0.266	0.556**	0.241	0.454***	0.148	0.480***	0.153
	MLE-based sequential VLR		Batch VLR		Batch - sequential VLR			
	$\hat{\beta}$	$\hat{\sigma}$	$\hat{\beta}$	$\hat{\sigma}$	$\hat{\beta}$	$\hat{\sigma}$		
constant	1.341**	0.587	1.545***	0.244	1.570***	0.244		
Relocation possbilities	-0.611**	0.273	-0.102	0.145	-0.083	0.146		
ICU capacity	0.874***	0.309	0.607***	0.159	0.629***	0.160		
Acute respiratory threat	-0.637**	0.306	0.171	0.160	0.175	0.161		
Age	-1.311***	0.325	-0.770***	0.155	-0.773***	0.156		
Comorbidity: Cognitive functioning	-0.659**	0.295	-0.541***	0.156	-0.560***	0.156		
Comorbidity: Cardiovascular	-0.549*	0.320	-0.607***	0.154	-0.621***	0.154		
Comorbidity: Pulmonary	-0.134	0.312	-0.376**	0.159	-0.392**	0.159		
Comorbidity: Renal	0.277	0.326	-0.372**	0.160	-0.384**	0.161		
Comorbidity: Liver	-0.137	0.309	-0.325**	0.157	-0.338**	0.158		
Comorbidity: Immunity	-0.196	0.376	-0.444***	0.163	-0.459***	0.164		
COVID pneumonia disease course	-0.409*	0.247	-0.324**	0.142	-0.315**	0.143		
BMI <20	0.121	0.282	-0.080	0.150	-0.094	0.151		
BMI >40	0.138	0.328	-0.078	0.153	-0.086	0.153		
Frailty	2.023***	0.403	1.010***	0.163	1.037***	0.164		
Patient decision	0.631**	0.265	0.450***	0.143	0.471***	0.144		

4.6.3 Model performance

Lastly, we discuss the performance of the models based on the COVID-19 data set. We have used the same manipulated K-fold cross-validation procedure as we have described for the special welfare case, with $K = 10$. For every fold, the real-life data set of 70 observations was split up into a training set of 47 observations and a test set of 23. Since there are 23 different patients in the real-life data set, every patient appears exactly once in the test set. The threshold for classification is again set to 0.5.

An important notion to keep in mind when analyzing the performance matrix is that the data set is severely imbalanced, as we have highlighted in Chapter 3. The real-life data set holds 54 positive decisions and only 16 negatives[7]. This imbalanced character also means that the training set is imbalanced. Therefore, the predictions are generally very high and, assuming a threshold of 0.5, a large fraction of the predictions are classified as positive.

In this data set, we detect two groups of approaches with an identical predictive performance: all MLE-related approaches and fully Bayesian approaches. Models within the same group make exactly the same classifications in every fold. We first focus on the group of fully Bayesian models. For none of these models the MCC performance metric could be computed in any of the folds. This can be explained by the imbalanced nature of the data set, which causes the models to not predict any negatives for any of the folds. Since all predictions are classified as positive, recall is equal to 1 and accuracy and precision are equal to the average fraction of positive decisions over all test sets.

On the other hand, there is the group of MLE-related models. These models perform substantially better in predicting negative decisions, since the MCC performance metric could be computed for 9 out of 10 folds. This means that this group of models was able to predict negative decisions for these 9 folds. Although predicting negative decisions, this group also classifies all true positive decisions correctly, which we can derive from the recall values being equal to 1. Moreover, the accuracy and precision are higher than the other group of models. We can thus state that the MLE-related approaches perform better than the fully Bayesian approaches based on the predictive performance metrics.

Furthermore, we also take the calibration error (CE) into account as a performance metric. We see that the average CEs are rather close to each other. However, the MLE-based approaches lead to a slightly lower CE than the fully Bayesian approaches. To get a better intuition of this difference, we have provided the calibration plots of the best and worst calibrated models in Figure 4.3. The worst calibrated model is a fully Bayesian model, for which predicted probabilities are centred around a high probability. This can be explained by low absolute posterior means and a high intercept term, which are caused by our choice for informative priors with low absolute means. The MLE logit model is allows for more variation in the predicted probabilities due to larger absolute posterior parameter values. We can conclude that the MLE-based approaches are better calibrated for this respective data set.

[7]We have considered applying bootstrapping approaches, such as oversampling, to balance the data set. Nevertheless, since Bayesian models are built on the assumption that the data set constitutes the entire population, bootstrapping data points is not allowed.

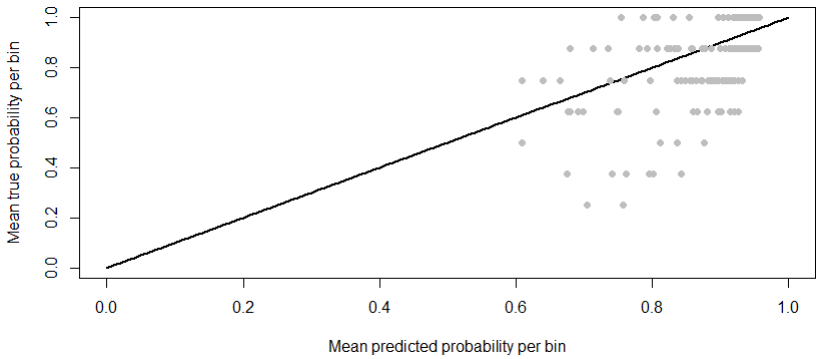
Table 4.9: Performances per model for the COVID-19 data set

Name model	Accuracy	Recall	Precision	MCC	Cal. error	Times
MLE probit	0.839 (10)	1 (10)	0.830 (10)	0.530 (9)	0.119 (3)	0:01
MLE-based sequential Gibbs	0.839 (10)	1 (10)	0.830 (10)	0.530 (9)	0.121 (2)	8:47
Batch Gibbs	0.774 (1)	1 (10)	0.774 (1)	NA (0)	0.129 (1)	0:16
Batch - sequential Gibbs	0.774 (1)	1 (10)	0.774 (1)	NA (0)	0.131 (0)	9:13
MLE logit	0.839 (10)	1 (10)	0.830 (10)	0.530 (9)	0.121 (2)	0:01
MLE-based sequential MH	0.839 (10)	1 (10)	0.830 (10)	0.530 (9)	0.123 (0)	26:13
Batch MH	0.774 (1)	1 (10)	0.774 (1)	NA (0)	0.130 (0)	0:43
Batch - sequential MH	0.774 (1)	1 (10)	0.774 (1)	NA (0)	0.131 (0)	26:29
MLE-based sequential VLR	0.839 (10)	1 (10)	0.830 (10)	0.530 (9)	0.123 (1)	0:01
Batch VLR	0.774 (1)	1 (10)	0.774 (1)	NA (0)	0.129 (1)	0:01
Batch - sequential VLR	0.774 (1)	1 (10)	0.774 (1)	NA (0)	0.131 (0)	0:01

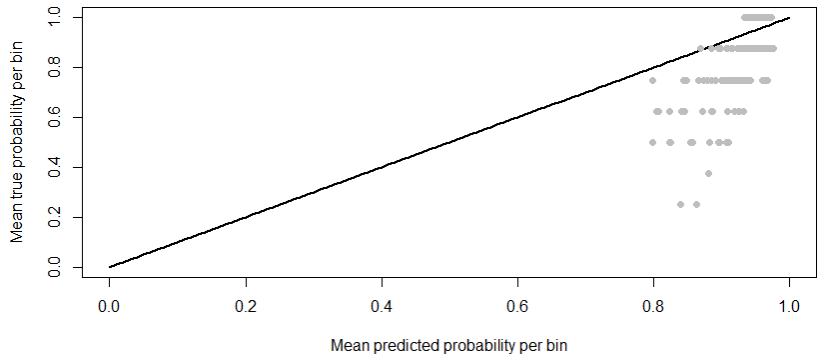
Notes: (i) The numbers in between brackets indicate for how many folds this model resulted in the optimal performance. (ii) Several entries of the MCC column contain value "*Not Available*" (NA), since the MCC could not be computed in any of the folds.

Figure 4.3: Calibration plots of the best and worst calibrated models for the COVID-19 data set

(a) Calibration plot of MLE logit



(b) Calibration plot of Batch - sequential Gibbs



4.7 Conclusion

A conclusion on which approach to BAIT can best be maintained should be reviewed on the five assessment criteria of Section 4.1. An approach should perform well on all criteria, since each of them is considered important for the proper functioning of BAIT and the satisfaction of end-users.

(i) *Matching model assumptions.* Bayesian methods are better suited for BAIT than classical methods for three reasons. Firstly, the interpretation of parameter estimates in Bayesian inference is more aligned with the reasoning behind BAIT. Secondly, a Bayesian approach to BAIT allows incorporating a priori knowledge of experts. Thirdly, Bayesian inference is better equipped for the small data sets of BAIT. When comparing MCMC methods Gibbs sampling and MH sampling to VLR, both have their advantages and disadvantages. MCMC methods are considered to be more exact than VLR. Nevertheless, VLR can be computed in real-time, whereas MCMC methods commonly suffer from long computation times.

(ii) *Suitable for sequential updating.* Bayesian inference is intrinsically better suited for the sequential updating of parameters than classical inference. In our approaches, the posterior density of the previous step was taken as the prior density in the next step. Our sequential approach potentially led to the problem of accumulating errors due to approximating the posterior by a Gaussian density at every step. This has proven not be a problem, since the *batch - sequential Bayesian* posteriors were very similar to the *batch Bayesian* posteriors.

(iii) *Good predictive performance.* In terms of predictive performance, no large differences between the assumed behavioural model or different Bayesian methods were detected. However, three groups of similarly performing approaches could be made based on the settings in which the modelling was applied: MLE approaches, MLE-based Bayesian approaches and fully Bayesian approaches. For a balanced data set, no straightforward conclusion could be drawn on the best performing approaches. There were no large deviations in accuracy between the models. Where recall was higher for MLE-related methods, the fully Bayesian approaches performed better on precision and MCC. For an imbalanced data set, the MLE-related approaches performed better than the fully Bayesian approaches. This better performance was caused by the fully Bayesian models only predicting positives for every fold.

(iv) *Good calibration.* For both data sets, MLE-related methods were better calibrated than fully Bayesian methods. We found this is mainly due to our choice for an informative prior with low absolute means, which caused low absolute posterior means of fully Bayesian methods.

(v) *Short computation time.* MLE and VLR can be calculated in real-time, whereas MCMC approaches require a substantial computation time. This is due to the simulations of the Markov chains. The computation time for batch approaches was considerably shorter than for sequential approaches. Moreover, the MH algorithm requires more computational effort than Gibbs sampling.

With respect to these assessment criteria, we consider *MLE-based sequential VLR* to be the most suitable model for the objectives of BAIT. As we have seen, a Bayesian model intrinsically suits the assumptions of BAIT better than a classical estimation routine and allows for sequential updating of parameters. Moreover, MLE-based Bayesian approaches make better calibrated predictions than the fully Bayesian methods and perform better on an unbalanced data set. Despite the approximation errors of VLR, our testing procedures did not indicate any large differences in predictive performance or calibration between VLR and MCMC methods. However, VLR has a much shorter computation time than MCMC methods, which is considered crucial to BAIT. Taking these arguments into account, we would recommend incorporating the *MLE-based sequential VLR* approach to BAIT.

Chapter 5

The feedback loop in dynamic BAIT

5.1 Introduction

Chapter 5 serves the following research goal:

2. Explore the consequences of the feedback loop of dynamic BAIT and evaluate how these consequences are impacted by altering feedback loop conditions;

In Section 5.2 we focus on gaining a better understanding of the feedback loop, based on related studies from other fields. In Section 5.3 the feedback loop is conceptualized for dynamic BAIT. This is followed by Section 5.4, in which we describe our simulation framework for exploring the consequences, define the investigated effects and for which conditions we investigate these effects. Section 5.5 then provides the results for our investigation. In Section 5.6 we conclude our findings for the feedback loop.

5.2 Literature review

Before diving into the conceptualization of the feedback loop of dynamic BAIT, we should first gain a broader understanding of what the term "feedback loop" actually entails. A feedback loop occurs when recommendations of BAIT are based on decisions made by experts that were exposed to previous recommendations of BAIT. Since feedback loops have not yet been a subject of research in the field of expert DSS, we are interested in how other fields view this phenomenon.

The feedback loop has been studied most extensively in the area of online recommendation systems. Abdollahpouri & Mansoury (2020) describe it as one of the root causes of popularity bias, which is one of the prevailing issues in these systems. Popularity bias is defined as the over-recommendation of a few popular items. The authors state that the initial occurrence of popularity bias is due to external biases, which means that some items are inherently more popular than others. Due to feedback loops, recommendation algorithms consequently have a higher tendency to recommend these items, which means that these particular items then also gain a larger number of user interactions. The popularity bias could thus be intensified over time when users interact with the given recommendations, that are biased towards popular items, and this interaction is added to the data. In another paper by the same authors, Mansoury et al. (2020) emphasize that in this way even a small bias in the current state of a recommendation system could be greatly amplified

over time if it is not addressed properly. When we translate these findings to the case of BAIT, this gives rise to the hypothesis that when a group of experts are more likely to make one particular decision, this preference can be amplified by multiple iterations through BAIT.

Jiang et al. (2019) have investigated degenerate feedback loops in recommendation systems. The authors define that a recommendation becomes "degenerate" when the interest of users becomes extreme. Degeneracy particularly occurs when the recommendation for a user is exactly equal to the user's interest, and can best be combated by also recommending other items than the one of interest or by increasing the number of items on the platform. In the case of BAIT, we do not consider expert-specific recommendations and we cannot increase the number of possible decisions. Nevertheless, we can learn that a strong alignment of recommendations with expert preferences might lead to extreme recommendations of BAIT, i.e. degeneracy.

In the long term, there are also other issues that might occur due to the feedback loop. Mansoury et al. (2020) have shown that the bias amplification in recommendation systems might lead to declining diversity of recommendations, shifting users preferences, and homogenization of user groups. These statements are supported by Chaney et al. (2018), who have found three main effects of feedback loop dynamics in online recommendation systems. These are homogenization of user behaviour, users experiencing losses in utility due to homogenization effects and an increasing inequality of item consumption. In the context of BAIT, these conclusions suggest that the feedback loop introduces the risk of homogenization of expert decisions and experts making suboptimal decisions. Both of these consequences are very undesirable. When expert decisions become homogeneous, this might lead to a fast determination of a decision, while the debate should actually still be continued. Moreover, making suboptimal decisions might have large effects, especially in the fields in which BAIT is currently applied, such as the medical sector.

However, not all recommendation system literature has found negative implications of the feedback loop. For example, Nguyen et al. (2014) are very positive about the use of movie recommendation systems at the individual level. The authors provide evidence that users who accept the advice of a personalized recommendation system, thus becoming part of the feedback loop, receive a more positive experience than users who do not. Moreover, they have found that the reduction in content diversity is relatively small and that recommendation-takers even consume more diverse movies. They state that the feedback loop might thus even play a broadening role. These findings are relatively contradictory to the previous findings, which implies that we cannot simply copy the findings of the online recommendation systems to dynamic BAIT. Therefore, we conduct our own research on the effect of dynamic BAIT on the homogenization of expert decisions.

On a final note, we shed some light on another field of study where the feedback loop is a subject of discussion. This field is "predictive policing": Given historical crime incident data, deciding how to allocate police officers to areas for crime detection (Lum & Isaac, 2016). In this field, a feedback loop occurs when models for predictive policing send officers to neighbourhoods with a high arrest count, thus leading to a larger arrest count in these neighbourhoods, which provides the data for updating the model. In this way, preexisting biases are further compounded. Ensign et al. (2018) state that this problem of "runaway feedback loops" is much less explicitly encountered when crime rates between regions are more or less equal. This gives rise to the thought that also in BAIT, the feedback loop becomes more of a problem when there is a larger preexisting tendency to one particular decision.

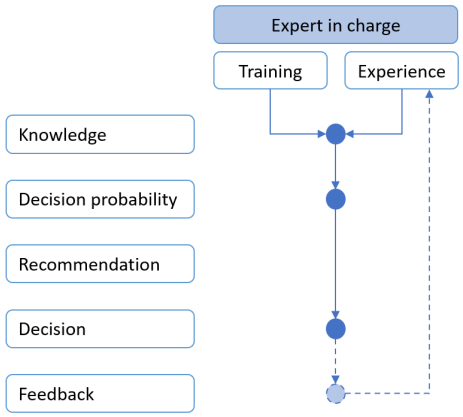
5.3 Conceptualizing the feedback loop

In order to investigate the feedback loop in dynamic BAIT, the loop itself should first be well-understood. The construction of the feedback loop has been inspired by the explanation of feedback loops between user behaviour and algorithmic recommendation systems (Chaney et al., 2018) and the architecture of a dynamic BAIT-based DSS (Schrama, 2021).

5.3.1 One expert, without BAIT

In order to arrive at the feedback loop of BAIT, we should first get insights on a simple feedback loop for expert decision-making. In Figure 5.1 we see how a single expert would make a decision without the involvement of BAIT. Following Larrick & Feiler (2015), we define an *expert* as a person who possesses domain-specific knowledge that is acquired through either *training* (exposure to knowledge) or *experience* (practice applying knowledge). When faced with a decision task, the expert bases his personal decision probability on the knowledge that he has. Without any external recommendations, the expert makes his decision based on his personal decision probability. The expert is able to effectively learn from his decisions and update his experience, once he receives clear and immediate feedback on his decision. In this particular case, *feedback* is defined as the objective outcome of a decision. The degree to which feedback is available largely depends on the kindness of the learning environment (Larrick & Feiler, 2015).

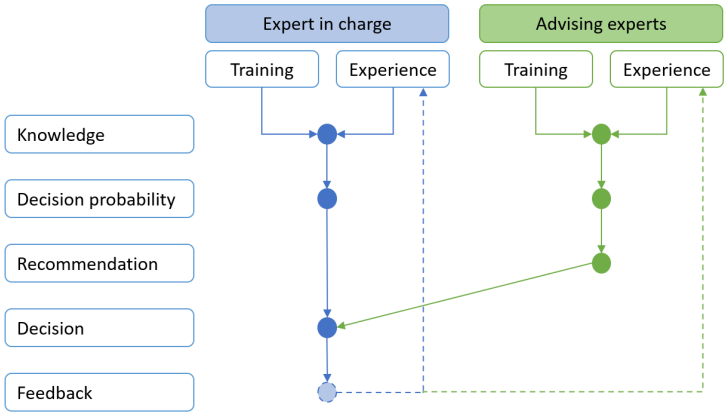
Figure 5.1: Feedback loop with an expert in charge and without BAIT



5.3.2 Multiple experts, without BAIT

In Figure 5.2, the simple feedback loop is extended to a version where other experts provide recommendations to the expert in charge of the decision. This feedback loop shows how decision-making in most settings is shaped without the interference of BAIT. Similar to the expert in charge, the advising experts also have their personal knowledge, from which they make a decision probability for a particular decision task. This decision probability gives rise to their recommendation to the expert in charge. The expert in charge can subsequently decide to which degree he takes the recommendation into consideration. When feedback on the choice is available, both the expert in charge and the recommending experts can learn from the feedback.

Figure 5.2: Feedback loop with an expert in charge, advising experts and without BAIT

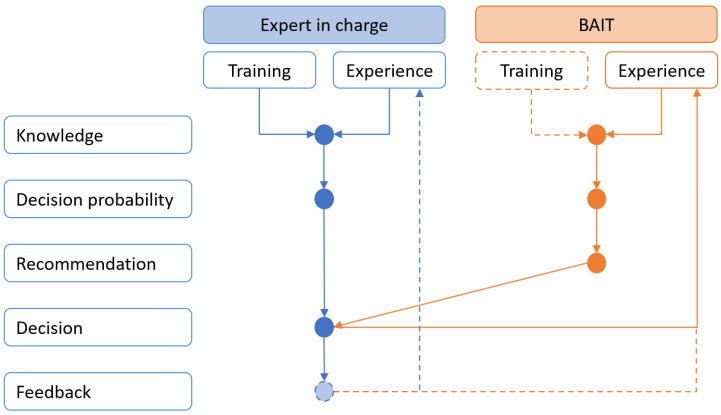


5.3.3 One expert, with BAIT

In Figure 5.3 we add BAIT to the framework, where we first consider the case with a single expert. The knowledge of BAIT consists of experience and potentially training. The experience comes from past decisions that experts have made, either in hypothetical or real-life scenarios, or possibly available objective feedback. Furthermore, in a Bayesian version of BAIT we would say that BAIT has received training when the prior beliefs of experts are included. Based on the knowledge of BAIT, the technology produces a decision probability, which is equivalent to the recommendation that it makes. Similar to the case with the advising experts, the expert decides the extent to which he takes this recommendation into account and subsequently makes a decision.

In a study on the moral autonomy of experts in BAIT, Yildiz (2021) argues that the degree to which BAITs recommendation affects the final decision depends on the level of automation of BAIT and the individual dependency of the expert in charge on BAIT. The level of automation states whether BAIT simply proposes a suggestion for a decision or automatically executes a decision. This level of automation is to be determined by the end-user of the technology. Ultimately, BAIT updates his experience by a decision a human makes for a particular decision task and uses this as extra knowledge for the next recommendation.

Figure 5.3: Feedback loop with an expert in charge and with BAIT

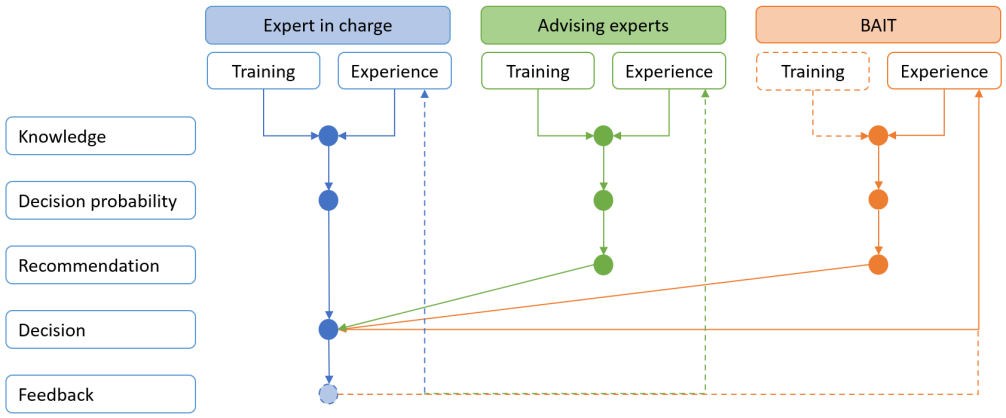


5.3.4 Multiple experts, with BAIT

In Figure 5.4 we see the complete feedback loop, where multiple experts and BAIT are involved. Both advising experts and BAIT make a recommendation based on their respective decision probabilities. The expert in charge makes the ultimate decision, based on his own decision probability and the recommendations he receives from both experts and BAIT. The degree to which BAITs recommendation affects the final decision depends on its level of automation and the individual dependency of the expert in charge on BAIT.

After a decision is made, it is used to update the experience and thus knowledge of BAIT. When feedback is available, experts and BAIT can learn and update their experience by this feedback. Nevertheless, we see here where BAIT is most effective: in cases where limited feedback is available and it is more difficult for experts to learn from the outcome of past decisions. BAIT can then offer a learning mechanism. For instance, it is hard to make an objective judgement on whether an intensivist has made the right decision when he did or did not admit a COVID-19 patient to the ICU. Then a recommendation of BAIT on what the colleagues of the intensivists would do in this situation could prove to be useful.

Figure 5.4: Feedback loop with an expert in charge, advising experts and with BAIT



5.4 Methodology

After yielding a clear picture of the existing literature on feedback loops and conceptualizing the feedback loop of dynamic BAIT, we can outline our research on the effects of this loop. As suggested by D'Amour et al. (2020), simulation is an essential element in studying the long-term dynamics of AI in decision-making processes. Hence, we also make use of simulation to investigate the effect of the feedback loops in dynamic BAIT.

In the following, we first describe on which assumptions our simulation is built (Section 5.4.1), which effects we investigate (Section 5.4.2) and for which conditions of the feedback loop we investigate these effects (Section 5.4.3).

5.4.1 Simulation assumptions

We make use of an empirically grounded simulation approach. Such an approach is defended by Kiesling et al. (2012), who state that conventional theoretical methods for studying long-term dynamics are often highly abstract. These methods are generally based on simple conceptions of human decision-making and do not aim to provide forecasts of managerial diagnostics for specific cases. The authors suggest that empirically grounded agent-based models provide better managerial guidance and policy analyses. Therefore, our simulation approach is based on empirical findings. We use the data set of the special welfare applications, which is described in Section 3.2. We start by simulating the beliefs of a group of N_{simexp} experts, after which we apply a number of simulations of decision tasks. These procedures are based on a series of assumptions.

Assumption 1: *The setting of dynamic BAITs feedback loop is without advising experts.*

The research needs to be demarcated. Its aim is to investigate the possible effects of a dynamic version of BAIT. Modelling the advice of other experts and investigating the extent to which the expert in charge takes their recommendation into account are not the focus of this study.

Assumption 2: *Expert knowledge is static over time.*

This assumption simplifies our model and prevents us from making any other structural assumptions. We have very limited knowledge of how experts change their beliefs over time. Experts might receive new training due to which their beliefs are updated or they might become increasingly or decreasingly confident on the use of BAIT. Since we have no information on this, we consider static expert beliefs a validated approach.

Assumption 3: *No objective feedback is available.*

BAIT thrives in situations where no objective feedback is available and it is hard to learn from past outcomes. It would therefore be an atypical case for BAIT if objective feedback was included.

Assumption 4: *The knowledge of BAIT only consists of experience*

In our conceptualization of BAITs feedback loop, the knowledge base of BAIT can be instigated by either training or experience. Experience can come from previously made decisions and objective feedback. We already assumed no objective feedback is available. Since the particular point of interest is researching the influence of past made decisions on BAIT, we will not haze the research by assuming any prior information to be provided to BAIT.

Assumption 5: *As dynamic version of BAIT, we apply the model MLE-based sequential VLR.*

MLE-based sequential VLR refers to the Bayesian approach where the prior is obtained by applying Maximum Likelihood Estimation on the experimental data and the posterior is updated iteratively by sequential Variational Logistic Regression. In Chapter 4, this method has proven to be suitable for sequential updating, to perform competitively and to be well calibrated. Moreover, VLR has a short computation time compared to the other Bayesian methods, which is beneficial for performing a large number of simulations in our research.

The experimental data set is yielded by using the vectors of individual beliefs $\boldsymbol{\beta}_1^{expert}, \ldots, \boldsymbol{\beta}_{N_{simexp}}^{expert}$ to determine the decisions the simulated experts would make on the 30 choice scenarios from the original data set. This experimental data set is used to estimate the initial model of BAIT. Hereafter, BAIT is updated after every expert decision i and time step t.

Assumption 6: *At every time step t, each expert i decides upon exactly one simulated decision task $\boldsymbol{x}_{it}$.*

In the real world, experts have varying work ethic and seniority, leading to differing numbers of decisions they complete. Nevertheless, determining which simulated experts to assign a greater number of decisions cannot be justified and would greatly impact the results of our simulation. This assumption simplifies our simulation and we believe that it allows us to capture the essence of the expert decision-making process.

Assumption 7: *Decision task matrix $X^{(sim)}$ is simulated according to Algorithm 6 provided in Appendix A.5.*

The algorithm takes the data type of all variables into account. Notice that we draw from uniform distributions, because we have no prior knowledge of the distribution of the data.

Assumption 8: *A large number M simulations are performed, where for every simulation the simulated decisions tasks $\boldsymbol{x}_{it}$ are presented in a different sequence.*

By reordering the sequence of the decision tasks, the impact of which decision task is presented to which expert at which time step diminishes.

Assumption 9: *For every decision task $\boldsymbol{x}_{it}$ at time t, BAIT and expert i have their respective decision probabilities P_{it}^{bait} and P_{it}^{expert} defined by:*

$$P_{it}^{bait} = G(\boldsymbol{x}_{it}'\boldsymbol{\beta}_{it}^{bait}) \tag{5.1}$$

$$P_{it}^{expert} = G(\boldsymbol{x}_{it}'\boldsymbol{\beta}_{i}^{expert}) \tag{5.2}$$

Since we make use of the VLR model, we assume the decision probabilities of experts follow a logistic distribution. $G(z)$ is used to denote the logistic CDF.

Assumption 10: *Final decision decision probability P_{it}^{final} follows from a weighted average between P_{it}^{bait} and P_{it}^{expert}:*

$$P_{it}^{final} = \lambda_{it}P_{it}^{bait} + (1-\lambda_{it})P_{it}^{expert} \tag{5.3}$$

We define expert dependency parameter $\lambda_{it} \in [0,1]$ as the extent to which expert i depends his final decision P_{it}^{final} on BAITs recommendation P_{it}^{bait} at time t.

Assumption 11: *The final decision expert i makes at time t is denoted by $\hat{y}_{it}$ and is based on decision threshold T, which we set default to 0.5:*

$$\hat{y}_{it} = \begin{cases} 1 & \text{if } P_{it}^{final} > T \\ 0 & \text{if } P_{it}^{final} \leq T \end{cases} \tag{5.4}$$

Assumption 12: *The vector of individual beliefs of experts are simulated from a normal distribution, defined by $\boldsymbol{\beta}_i^{expert} \sim N(\boldsymbol{\beta}^{MLE}, \kappa\Sigma^{MLE})$, according to Algorithm 7 in Appendix A.6.*

Similar to Chapter 4.4, we assume a normal distribution of expert beliefs on parameters. We base the parameter values on MLE applied on the experimental dataset. We take the location parameters as the vector of MLE coefficients $\boldsymbol{\beta}^{MLE}$. Determining the heterogeneity between experts is more complex. Although Σ^{MLE} reflects the relative uncertainty in parameters, the entries of this

covariance matrix are too small to describe the heterogeneity between experts. Hence, we scale Σ^{MLE} by scaling factor κ.

The scaling factor is set according to the metric we apply for measuring the initial heterogeneity between experts, which is their initial agreement rate[1,2]. Let us denote the initial agreement rate on the experimental data set of two experts r and s by $AR_0(r,s)$. The decisions these two experts make on the experimental data set are then given by $\boldsymbol{y}_{r,0}$ and $\boldsymbol{y}_{s,0}$ respectively. Denoting the number of hypothetical decision tasks in the experimental data set by D_{hyp}, AR_0 is defined as:

$$AR_0(r,s) = \frac{1}{D_{hyp}} \sum_{l=1}^{D_{hyp}} \mathbf{1}\{y_{r,0,l} = y_{s,0,l}\} \tag{5.5}$$

Using this definition, and assuming we simulate N_{simexp} experts, we can derive mean initial agreement rate $\overline{AR}_0$:

$$\overline{AR}_0 = \frac{1}{\frac{1}{2}N_{simexp}(N_{simexp}-1)} \sum_{r=1}^{N_{simexp}-1} \sum_{s=r+1}^{N_{simexp}} AR_0(r,s) \tag{5.6}$$

Scaling factor κ is set such that $\overline{AR}_0$ between the simulated experts is approximately as large as $\overline{AR}_0$ between the experts in the actual data set. Expert belief vectors $\boldsymbol{\beta}_1^{expert}, \ldots, \boldsymbol{\beta}_{N_{simexp}}^{expert}$ are eventually simulated from $\boldsymbol{\beta}_i^{expert} \sim N(\boldsymbol{\beta}^{MLE}, \kappa\Sigma^{MLE})$. The complete methodology for the simulation can be found in Algorithm 7 in Appendix A.6.

5.4.2 The investigated effects

After having clarified our simulation procedures, we provide an introduction of the investigated effects. The literature research specifically suggests two main effects that feedback loops in recommendation systems might have: the amplification of recommendations due to preexisting user biases and homogenization of user behaviour. We transfer these to the two effects of the feedback loop of BAIT that we find most relevant to investigate.

Effect 1: *Amplification of BAITs recommendations*

We say a recommendation is amplified when BAITs recommendation for a particular case moves towards extremes over time. We explore under which conditions BAITs recommendations are amplified, what the magnitude of this amplification is and what the effect of such an amplification is on the final decision probabilities of experts.

We explore the amplification of BAITs recommendation in the following way. At every time step t, we present the simulated experts with some test case $\tilde{\boldsymbol{x}}_k$ and track $\tilde{P}_{t,k}^{bait}$, BAITs recommendation with respect to this test case. We evaluate BAITs recommendation amplification by tracking the

[1]The agreement rate metric is inspired by Chaney et al. (2018), who paired users based on their similarity in preferences regarding item consumption in online recommendation systems. The authors compared the Jaccard index of the paired users' interactions with a set of items against the Jaccard index of the same users when they were exposed to recommendations.

[2]The agreement rate has strong ties to the average squared Euclidean distance between expert decisions, which is a proper heterogeneity metric according to Tang et al. (2021). For the initial agreement rate we have that $AR_0(r^*,s^*) = \frac{1}{D_{hyp}}\sum_{l=1}^{D_{hyp}} \mathbf{1}\{y_{r^*0,l} = y_{s^*0,l}\} = 1 - \frac{1}{D_{hyp}}\sum_{l=1}^{D_{hyp}} \|y_{r^*0,l} - y_{s^*0,l}\|^2$

evolution of BAITs recommendation $\Delta\tilde{P}^{bait}_{t,k}$, which is defined as:

$$\Delta\tilde{P}^{bait}_{t,k} = \tilde{P}^{bait}_{t,k} - \tilde{P}^{bait}_{0,k} \tag{5.7}$$

We expect that the magnitude of $\Delta\tilde{P}^{bait}_{t,k}$ depends on the magnitude of $\tilde{P}^{bait}_{0,k}$, the initial recommendation of BAIT. We investigate this by tracking the evolution of recommendations for test cases with different initial recommendations. We find these test cases in the following way. A large number of N_{test} test cases are generated according to Algorithm 6 provided in Appendix A.5. These test cases form matrix $\tilde{X} = \{\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_{N_{test}}\}$. Let us denote c as the desired initial recommendation of a scenario. Then we select test case $\tilde{\boldsymbol{x}}_k$ such that $\tilde{P}^{bait}_{0,k} \approx c$.

Effect 2: *Homogenization of expert behaviour*

Homogenization of user behaviour is a frequently addressed risk of continuous iterations through the feedback loop in the recommendation system literature. Therefore, we explore the effect of feedback loops on the homogenization of expert behaviour due to the dynamic character of BAIT. Homogenization of expert behaviour is explored by tracking the evolution of the agreement rate between the two experts with the most heterogeneous beliefs. We believe this is a tangible metric for evaluating the homogenization effects of the feedback loop of dynamic BAIT.

Let us denote the initial agreement rate between experts r and s by $AR_0(r, s)$, as defined in Equation 5.5. Then we identify the two most heterogeneous experts by finding the two experts r^* and s^* who have the lowest initial agreement rate:

$$\{r^*, s^*\} = \arg\min_{r,s} AR_0(r, s) \tag{5.8}$$

Hereafter, we evaluate whether the decision these experts take become increasingly similar over time due to the involvement of BAIT. This is done by studying the evolution of AR_t. At every time step t one new decision task $\check{\boldsymbol{x}}_t$ is generated according to Algorithm 6 provided in Appendix A.5. Similar to Assumption 8, the impact of which particular decision task is presented at which time step is reduced by reordering the sequence in which the decision tasks are presented to the experts. This reordering is done for a large number M simulations.

We present the two heterogeneous experts the test case and compute which final decisions $\check{y}_{r^*t}$ and $\check{y}_{s^*t}$ they make. At time t we define the set of BAIT-influenced decisions as $\check{\boldsymbol{y}}_i = \{\check{y}_{i1}, \ldots, \check{y}_{it}\}$. The agreement rate at time t is then given by:

$$AR_t(r^*, s^*) = \frac{1}{D_{hyp} + t} \left(\sum_{l=1}^{D_{hyp}} \mathbf{1}\{y_{r^*0,l} = y_{s^*0,l}\} + \sum_{m=1}^{t} \mathbf{1}\{\check{y}_{r^*m} = \check{y}_{s^*m}\} \right) \tag{5.9}$$

The change in agreement rate is eventually defined as:

$$\Delta AR_t(r^*, s^*) = AR_t(r^*, s^*) - AR_0(r^*, s^*) \tag{5.10}$$

5.4.3 The investigated conditions

It is specifically of interest to know under which conditions these feedback loop effects are increasingly likely to occur. We have identified four conditions of which we find it most interesting to

investigate the consequences of the feedback loop effects. We assume the existence of one baseline scenario that embodies a likely scenario with typical conditions of the feedback loop. In this baseline scenario, a dynamic version of BAIT is implemented, with expert dependency λ_{it} based on the certainty of BAIT and a group of 10 experts with a regular heterogeneity. Based on this baseline scenario, we deviate one of these conditions at a time, ceteris paribus, to investigate the impact of that particular condition.

Condition 1: *The stage of implementation of BAIT*

In the baseline scenario, we assume a dynamic version of BAIT, where BAIT is updated at every new decision. The baseline scenario is compared to two alternative scenarios. First, we consider a scenario where a static version of BAIT is implemented, which means that BAIT is only estimated once, based on the hypothetical choice scenarios. The second alternative is a scenario without implementation of BAIT, where experts only base their choice on their own decision probability.

Condition 2: *Expert dependency on BAIT*

We explore the expert dependency by parameter λ_{it}, which we defined in Equation 5.3: $P_{it}^{final} = \lambda_{it}P_{it}^{bait} + (1-\lambda_{it})P_{it}^{expert}$. In the baseline scenario the magnitude of the expert dependency on BAIT linearly depends on the certainty of BAIT. We define this linear dependency as $\lambda_{it} = 2 \cdot |P_{it}^{bait} - 0.5|$.

We consider three alternative scenarios. The first is a "fully automated" version of BAIT, where $\lambda_{it} = 1$ for all i and t. Secondly, we consider a "fifty-fifty dependence" situation, where $\lambda_{it} = 0.5$ for all i and t. This means that the final decision probability is based for 50% on BAITs recommendation and 50% on the expert's decision probability. The last alternative we label as a "automation at threshold" situation. In this situation experts are autonomous, but BAITs recommendations are automated above 80% or below 20%: $\lambda_{it} = \mathbf{1}\{P_{it}^{bait} \leq 0.2 \text{ or } P_{it}^{bait} \geq 0.8\}$.

Condition 3: *Expert heterogeneity*

As a third condition, we study how severely heterogeneity between experts impacts the evolution of BAITs recommendation. We investigate this heterogeneity by applying variations to the individual expert beliefs. Here we consider five scenarios. In the baseline scenario, all experts follow the beliefs as defined in Assumption 12:

$$\boldsymbol{\beta}_i^{expert} \sim N(\boldsymbol{\beta}^{MLE}, \kappa\Sigma^{MLE}) \tag{5.11}$$

First, we compare this baseline scenario to two scenario's where experts are either more homogeneous or heterogeneous. Experts in these scenarios are simulated from a similar normal distribution as in Equation 5.11, but with a different scaling factor. These scaling factors are denoted by respectively κ_{hom} and κ_{het}. Similar to the baseline scenario, we yield these scaling factors according to some desired average agreement rate $\overline{AR}_0$. For a homogeneous group of experts, the desired $\overline{AR}_0$ should be chosen higher than $\overline{AR}_0$ in the actual data set, and for the heterogeneous group, the value should be lower. Using these values, κ_{hom} and κ_{het} are determined using Algorithm 7 in Appendix A.6.

Moreover, we also consider two scenarios where there are experts with opposing views. In the first scenario, there is one opposing expert and $N_{simexp} - 1$ regular experts. In the second scenario, there is a group of opposing experts and a group of regular experts, which both have size $\frac{1}{2}N_{simexp}$.

Opposing experts are simulated according to:

$$\boldsymbol{\beta}_i^{expert} \sim N(-\boldsymbol{\beta}^{MLE}, \kappa\Sigma^{MLE}) \tag{5.12}$$

Condition 4: *Group size*

For the last condition, we investigate how the size of the group of experts N_{simexp} influences the evolution of BAITs recommendation and the homogeneity of experts. In the baseline scenario, we assume $N_{simexp} = 10$. We study how this baseline scenario relates to a scenario where there is a small number of experts ($N_{simexp} = 3$) and a large number of experts ($N_{simexp} = 25$).

5.5 Results

This section presents the results of the feedback loop investigation. In section 5.5.2 we study the evolution of two investigated effects for the baseline scenario. After this, Sections 5.5.3 to 5.5.6 provide the simulation results of BAITs recommendation amplification and expert homogenization for each of the investigated conditions.

5.5.1 Simulated experts and test cases

For the baseline experts, we computed a scaling factor κ of the MLE covariance matrix according to the average initial agreement rate in the actual data set, which is equal to 0.749. By taking this as the goal for the average initial agreement rate of regular experts, we came to κ equal to 6.5 and an average initial agreement rate of 0.742. The goal of the average initial agreement rate was set to 0.9 for the homogeneous group of experts and to 0.5 for the heterogeneous group. This has led to respective scaling factors κ_{hom} equal to 1.9 and κ_{het} equal to 51. The individual beliefs of the simulated experts can be found in Appendix D. In the following, experts 101 to 125 refer to regular experts, experts 201 to 205 to opposing experts, experts 301 to 310 to homogeneous experts and 401 to 410 to heterogeneous experts.

Moreover, we simulate a large number of possible progressions of BAIT, which is done by reordering the decision task matrix $X^{(sim)}$. During these simulations, we also vary in which decision tasks are presented to experts for evaluating their agreement rate. The number of simulations is set to $M = 100$.

For investigating recommendation amplification, it is crucial to note the impact of which particular test case we are reviewing. In particular, we should account for the magnitude of $\tilde{P}_0^{bait}$, BAITs initial recommendation for the test case. For example, a weak initial recommendation of 55% might lead to a different amplification than a strong initial recommendation of 95%. Therefore, in the investigation of BAITs recommendation amplification, we review five different test cases with $\tilde{P}_0^{bait}$ approximately equal to 55%, 65%, 75%, 85% and 95%.[3] An important caveat is that under some conditions, no test cases with large $\tilde{P}_0^{bait}$ are found, which is caused by small absolute initial parameter estimates. Therefore, the recommendation amplification for these test cases could not be evaluated.

[3]We only investigate the amplification of positive recommendations (i.e. $\tilde{P}_0^{bait} > 0.5$) towards upper bound 1. Amplification of negative recommendations towards lower bound 0 are not considered, since we assume similar amplification trends would occur.

The results of the evolution of BAITs recommendation $\tilde{P}_t^{bait}$ for these test cases for all investigated conditions can be found in Table 5.1. The evolution of the agreement rates between the two most heterogeneous experts AR_t for all investigated conditions are displayed in Table 5.2. In these tables we have chosen to display $\Delta\tilde{P}_t^{bait}$ and ΔAR_t for $t = 20$ and $t = 100$, since we consider these time points to be indicative for respectively the short and the long term.

5.5.2 Baseline scenario

We start by yielding a clear picture of how $\tilde{P}_t^{bait}$ evolves over time for each of the test cases for the baseline scenario. The baseline scenario focuses on a setting where there are 10 regular experts, for which we take simulated experts 101 to 110. Moreover, in the baseline scenario, we assume a dynamic version of BAIT, with expert dependency λ_{it} based on the certainty of BAIT.

The first result we retrieved is that no test case could be found such that $\tilde{P}_0^{bait}$ is approximately equal to 0.95. This points to relatively small absolute parameter values at time $t = 0$. The evolution of $\Delta\tilde{P}_t^{bait}$ for the other test cases is given in Figure 5.5. In this figure, we see that BAITs recommendations increase substantially in the short term, but reach a more steady state in the long term. This can be explained by the increasing amount of data, which causes a lower variation in parameters over time.

For all test cases, the average recommendations over all simulations are significantly amplified. The smallest change in recommendation is for the 85% case, which still increases by 10% in the long term. That this recommendation is amplified the least can obviously be explained by the initial recommendation being closer to asymptotic boundary of 100%. The largest increase occurs for the 65% case, which increases by more than 20% in the long term. Interestingly, the initial probability of 65% increases much more than the one of 55%. This implies that the smallest initial recommendations are not necessarily amplified the most.

Furthermore, we explore the agreement rate (AR) between the two most heterogeneous experts. Based on the initial agreement rate, simulated experts 109 and 110 have the most heterogeneous individual beliefs. The AR between these experts on the experimental data set is 0.533, which means they agreed on 16 of the 30 hypothetical scenarios. In Figure 5.6 we can see how the AR evolves over time for the baseline scenario. The plot shows that the AR between the two most heterogeneous experts increases to almost 80% over time. Their decisions thus become increasingly homogeneous.

Figure 5.5: Evolution of $\Delta\tilde{P}_t^{bait}$ for different values $\tilde{P}_0^{bait}$ in the baseline scenario

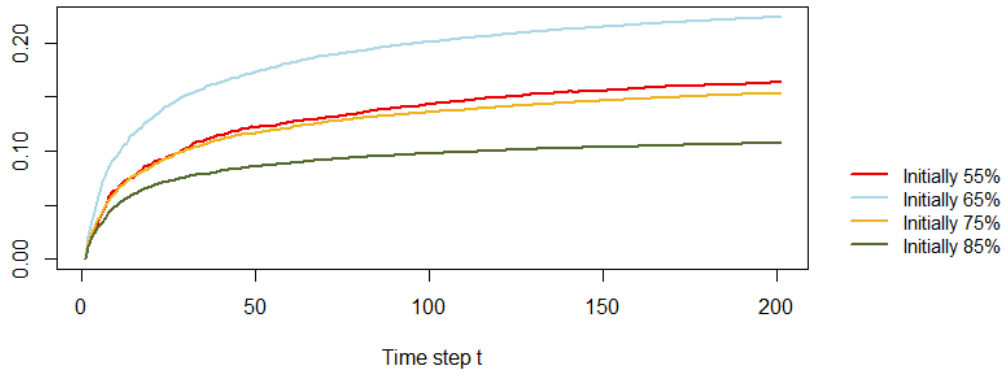


Figure 5.6: Evolution of $AR_t(r^*, s^*)$ in the baseline scenario

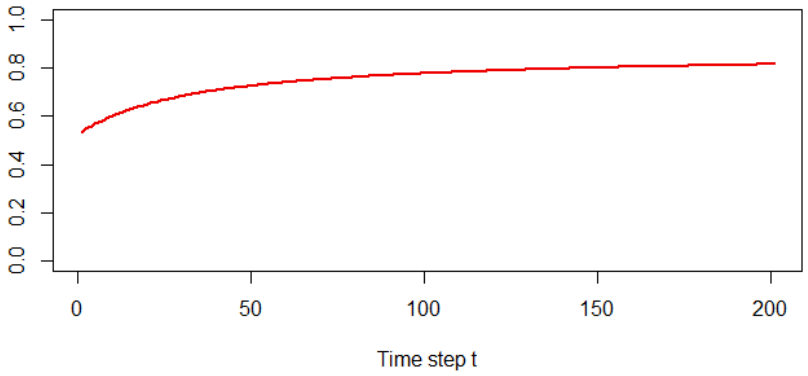


Table 5.1: Evolution of BAITs recommendation $\tilde{P}_t^{bait}$

		$\tilde{P}_0^{bait} \approx 0.55$		$\tilde{P}_0^{bait} \approx 0.65$		$\tilde{P}_0^{bait} \approx 0.75$		$\tilde{P}_0^{bait} \approx 0.85$		$\tilde{P}_0^{bait} = 0.95$	
		$\Delta\tilde{P}_{20}^{bait}$	$\Delta\tilde{P}_{100}^{bait}$	$\Delta\tilde{P}_{20}^{bait}$	$\Delta\tilde{P}_{100}^{bait}$	$\Delta\tilde{P}_{20}^{bait}$	$\Delta\tilde{P}_{100}^{bait}$	$\Delta\tilde{P}_{20}^{bait}$	$\Delta\tilde{P}_{100}^{bait}$	$\Delta\tilde{P}_{20}^{bait}$	$\Delta\tilde{P}_{100}^{bait}$
	Baseline scenario	0.09	0.144	0.132	0.201	0.088	0.136	0.069	0.098	NA	NA
Stage of implementation	Static BAIT	0	0	0	0	0	0	0	0	NA	NA
	No BAIT	0	0	0	0	0	0	0	0	NA	NA
Expert dependency	Fifty-fifty dependence	0.074	0.093	0.112	0.162	0.072	0.102	0.06	0.08	NA	NA
	Fully automated	0.065	0.11	0.074	0.113	0.062	0.088	0.037	0.042	NA	NA
	Automation at threshold	0.103	0.167	0.154	0.233	0.095	0.15	0.066	0.097	NA	NA
Expert heterogeneity	Heterogeneous	0.023	0.054	0.073	0.162	0.073	0.149	NA	NA	NA	NA
	Homogeneous	0.014	-0.009	0.064	0.096	0.061	0.107	0.037	0.05	0.023	0.034
	One outsider	0.118	0.16	NA	NA	NA	NA	NA	NA	NA	NA
	Two opposing groups	-0.013	-0.012	NA	NA	NA	NA	NA	NA	NA	NA
Group size	Small group	0.102	0.146	0.027	0.018	0.028	-0.008	0.062	0.091	0.026	0.032
	Large group	0.019	0.028	0.127	0.198	0.084	0.11	0.061	0.084	0.029	0.037

Notes: Values of $\Delta\tilde{P}_t^{bait}$ are NA (Not Available) when no test case with $\tilde{P}_0^{bait} \approx c$ was found.

Table 5.2: Evolution of agreement rate of the two most heterogeneous experts $AR_t(r^*, s^*)$

		AR_0	ΔAR_{20}	ΔAR_{100}
	Baseline scenario	0.533	0.121	0.243
Stage of implementation	Static BAIT	0.533	0.109	0.211
	No BAIT	0.533	0.034	0.064
Expert dependency	Fifty-fifty dependence	0.533	0.109	0.22
	Fully automated	0.533	0.187	0.359
	Automation at threshold	0.533	0.091	0.185
Expert heterogeneity	Heterogeneous	0.167	0.076	0.223
	Homogeneous	0.8	0.073	0.142
	One outsider	0.067	0.253	0.498
	Two opposing groups	0.067	0.008	0.019
Group size	Small group	0.667	0.073	0.149
	Large group	0.533	0.107	0.213

5.5.3 Stage of implementation

The first condition to review is the stage of implementation of BAIT. We explore what the effects are if we implement dynamic BAIT versus when we implement static BAIT or no BAIT. From these scenarios, dynamic BAIT is the only one that allows for learning over time. Hence, for both static BAIT and no BAIT, BAITs recommendations are not amplified over time.

More interesting is to explore how homogenization evolves in each of these settings of BAIT. In each of the scenarios, the same group of experts is taken. Hence, simulated experts 109 and 110 have the most heterogeneous individual beliefs with an initial AR of 0.533. The long term AR in the scenario without implementation of BAIT gives us a good indication of the true AR between the experts. That the AR in this scenario slightly increases in the long term implies that the initial AR was slightly lower than the true AR. The results show that an implementation of either static or dynamic BAIT causes a significant increase in AR. Nevertheless, this increase is a bit higher for dynamic BAIT than for static BAIT.

5.5.4 Expert dependency

The impact of the feedback loop is strongly related to the assumptions made on λ_{it}, which states the extent to which an expert depends his decision on the recommendation of BAIT. Therefore, this is the next condition we review. One by one, we explore three alternatives to the baseline scenario, where linear dependence of λ_{it} on the certainty of BAIT is assumed.

The first alternative is a "fifty-fifty dependence" scenario, where $\lambda_{it} = 0.5$ for every i and t. In general, BAITs recommendations are substantially amplified, but not as much as in the baseline case. In the baseline case, the experts rely stronger on BAITs recommendations as they become more extreme. For this reason, BAIT is reinforced to make more extreme recommendations. In the "fifty-fifty dependence" scenario, this reinforcement does not take place and the amplification is thus smaller. Furthermore, the increase in the agreement rate between the two most heterogeneous experts of 22% can be perceived as considerable. Nevertheless, this increase is not as large as in the baseline case, where the amplified recommendations lead to a larger impact of BAIT on the final decision and thus a stronger homogenization.

Next, we consider a scenario with full automation of BAITs recommendation, i.e. $\lambda_{it} = 1$ for every i and t. Based on our simulations, full automation of BAIT leads to a lower recommendation amplification than in other scenarios. This can be explained by BAIT only being exposed to its own recommendations and not to expert behaviour, which means that the technology stops learning. Automation of BAIT does lead to an enormous increase in AR. This is trivial, since experts will make exactly the same decisions for every decision task from the moment BAIT is activated.

The last alternative scenario is the scenario where BAITs recommendations are automated when they reach a threshold, which is mathematically denoted as $\lambda_{it} = \mathbf{1}\{P_{it}^{bait} \leq 0.2 \text{ or } P_{it}^{bait} \geq 0.8\}$. This scenario leads to the largest increase in recommendations of all alternative scenarios. This is due to the fact that recommendations of BAIT are only followed in case the threshold values are reached. In this way, BAIT is strongly reinforced to make extreme recommendations. The homogenization effects for this scenario are smaller, since BAIT only homogenizes expert decisions in case recommendations reach the threshold values. If the threshold values are not reached, expert decisions are not affected by BAIT at all.

5.5.5 Expert heterogeneity

This section explores the evolution of BAIT for four scenarios with a different expert heterogeneity than in the baseline scenario. The expert heterogeneity in the baseline scenario is derived from the heterogeneity of experts in the actual data set. A striking initial note is that for the more hetero-

geneous or polarised scenarios no large initial recommendations could be found. The conservative initial recommendations are due to small absolute parameter values in the initial model estimation. We suspect this is caused by the large diversity in decisions made by heterogeneous or polarised groups of experts.

The first alternative is a scenario where experts are heterogeneous. We consider simulated experts 401 to 410. For this group of experts, values of $\tilde{P}_0^{bait}$ until 75% were found. The recommendations for the available test cases are amplified over time. The smallest initial recommendation increases only slightly, but the other two test cases undergo a larger increase. Furthermore, simulated experts 401 and 402 are identified as the most heterogeneous experts. These experts initially agree on only 16.7% of the hypothetical choice scenarios. We see that due to the interference of BAIT, the AR between these experts increases, although it does not exceed 40%.

Secondly, we consider a more homogeneous group of experts, consisting of simulated experts 301 to 310. BAIT is able to predict extreme probabilities for this group of experts, but the amplification of these recommendations is smaller than in previous scenarios. The smallest initial recommendation is not amplified, but even slightly reduces over time. The larger initial recommendations are amplified. The most heterogeneous experts in this group are simulated experts 302 and 306. The already high initial AR of 80% between these experts increases over time to 94%. Hence, we can argue that when the group of experts is homogeneous, BAIT helps to reduce the remaining disagreement between experts.

The two scenarios with outsiders both result in conservative initial recommendations of BAIT that do not exceed 60% for any decision task. Hence, only one test case is available for these scenarios. For the scenario with one outsider, we consider simulated expert 201 as an outsider and simulated experts 101 to 109 as experts with regular beliefs. The recommendation of the only available test case is substantially amplified by 16%. Moreover, BAIT leads to an enormous increase in homogeneity between the outsider and his biggest opponent, which is simulated expert 101. They initially agree on only 6.7% of the cases. Due to the inference of BAIT, their AR increases to roughly 25% in the short term and 50% in the long term.

In the last scenario, there are two equally sized groups of opposing experts. The group with regular belief consists of simulated experts 101 to 105 and the group with opposing belief consists of simulated experts 201 to 205. The extreme heterogeneity in beliefs causes initial recommendation $\tilde{P}_0^{bait} \approx 55\%$ to not be amplified, but reduced over time. BAITs recommendations thus stay within the conservative margins. The two most heterogeneous experts from the two groups are simulated experts 101 and 201, the same as in the previous scenario. Unlike the scenario with one outsider, BAIT does not have the power to bridge the disagreements between the experts, since the low initial AR only increases slightly due to BAIT.

5.5.6 Group size

The last condition we investigate is varying group size. For the small group size, we consider simulated experts 101 to 103 and for the large group size simulated experts 101 to 125. These are compared to the baseline scenario of 10 experts.

In case of a small group of experts, BAITs recommendations do not move in one particular direction. Some recommendations are amplified, while others are slightly reduced. This lack of movement in one direction can be attributed to the small data size, which leads to a larger variation

in parameter estimates during the process of sequential updating. The two most heterogeneous experts are simulated experts 102 and 103, which have a rather high initial AR. This AR is amplified over time due to BAIT.

For the large group of experts all recommendations are amplified, but with differing magnitudes. The initial recommendations of 55% increases by only 3%, while the initial recommendation of 65% increases by almost 20%. We see that the short term amplification is very indicative of the long term amplification. This is caused by the large amount of data that is already available in the short term, which enables a steady evolution of the influence of variables. The two experts with the most heterogeneous beliefs are simulated experts 107 and 122. Due to the large amount of data and resulting high level of certainty of BAIT, the decisions of these experts become considerably more homogeneous.

5.6 Conclusion

In this chapter, we have followed a systematic approach to achieve our research objective. Whereas reviewing existing literature helped us to identify which effects to study, conceptualizing the feedback loop enabled us to determine our simulation assumptions and the most crucial conditions to investigate. In this section, we draw our conclusions to which extent the two investigated effects occur in BAITs feedback loop and how altering conditions might influence these effects.

Effect 1: *Amplification of BAITs recommendations*

A recommendation of BAIT for a particular decision task is said to be amplified when the recommendation moves towards extremes over time. Based on our simulations, BAITs recommendations are likely to be substantially amplified due to continuous iterations through the feedback loop. Assuming an expert dependency based on the certainty of BAIT intensifies this amplification, since experts depend stronger on BAIT in case more extreme recommendations are made. BAIT is thus reinforced to make extreme recommendations. In case of automation from a certain threshold, this reinforcement of extreme recommendations is even more apparent, which is why this scenario leads to an even larger amplification. Moreover, full automation of recommendations leads to a lower amplification, since BAIT only learns from its own recommendations and not from new expert knowledge.

In our simulations, we found a rather large effect of the composition of the group of experts on the evolution of BAITs recommendations. Interestingly, BAIT was initially not able to provide any large recommendations when experts with opposing views were included in the simulations. This issue appears to resolve itself over time in case of only one outsider, but not when two groups of opposing experts are considered. In comparison to the scenario of a heterogeneous group of experts, BAIT was able to provide more extreme initial recommendations in case of a homogeneous group of experts. Nevertheless, the recommendations are more severely amplified in case of a heterogeneous group compared to a homogeneous group. Furthermore, we found that group size mainly influences the long-term movement of recommendations. For a small group of experts, BAITs short-term recommendations might be inconsistent with the long-term recommendations. For a large group of experts, BAITs short-term recommendations are very indicative for the long-term.

Effect 2: *Homogenization of expert behaviour*

In this study, we say expert behaviour homogenizes when the agreement rate (AR) between the two most heterogeneous experts increases over time. Based on our simulations, implementing either a static or a dynamic version of BAIT leads to substantial homogenization of expert behaviour, albeit dynamic BAIT to a higher extent. In the baseline scenario, we see that the combination of amplified recommendations and a stronger expert dependency when recommendations become more extreme leads to a large homogenization of expert behaviour. The largest increase in AR is achieved by fully automating BAITs recommendations, since expert decisions are fully homogenized. Automation from a threshold value leads to smaller homogenization effects.

Moreover, our simulations suggest that when we assume a heterogeneous group of experts with a lower initial AR, BAIT can lead to a serious increase in homogeneity. Even for a homogeneous group of experts, BAIT stimulates the already high initial AR to increase even further. In case of the occurrence of experts with opposing opinions, we have seen that the number of opposing experts plays a significant role. When only one opposing expert occurs, BAIT is able to harmonize the decisions of this opposing expert with the rest of the group. When two equally sized opposing groups occur, the segregated decision pattern persists nonetheless. On a final note, our simulations do not provide any evidence that the size of the group of experts has a large impact on the homogenization effects.

Chapter 6

Conclusion

The conclusion of this study is divided into four sections. It starts by stating the main findings in Section 6.1. This is followed by summarizing the main contributions in Section 6.2. Hereafter, Section 6.3 discusses the limitations and recommendations for future research. The chapter is concluded by our final recommendations to Councyl in Section 6.4.

6.1 Main findings

This paper has served two main research objectives:

1. Design several Bayesian alternatives to BAIT and assess which model is most suitable for the objectives of BAIT;

2. Explore the consequences of the feedback loop of dynamic BAIT and evaluate how these consequences are impacted by altering feedback loop conditions;

In order to reach the first objective, we have formulated nine Bayesian approaches. All of these approaches have varying assumptions on the prior density, the incorporation of sequential updating, the behavioural model and the Bayesian method. These models were compared to each other and their MLE counterparts based on a set of five assessment criteria. Based on our analysis, we came to the conclusion that from the considered models *MLE-based sequential Variational Logistic Regression (VLR)* is the most suitable for the purposes of BAIT. This consideration is based on Bayesian models intrinsically being more suitable to BAIT, VLR being appropriate for sequential and real-time updating and MLE-based models being well calibrated and having a proper predictive performance on imbalanced data sets.

With regard to the second research objective, we have considered two effects of the feedback loop of dynamic BAIT, which are BAITs recommendation amplification and homogenization of expert behaviour. Our simulations have indicated that BAITs recommendations are substantially amplified by its feedback loop. This amplification is reduced by considering a homogeneous group of experts. Automating decisions leads to ambiguous amplification effects. Whereas recommendations are not significantly amplified in a full automation setting due to the lack of new expert knowledge, they are strongly pushed towards threshold values in case of automation of BAIT from a threshold value. Furthermore, our simulations provide evidence for stating that any implementation of BAIT

leads to some extent of homogenizing expert behaviour. The homogenization effects are strong in our assumed baseline scenario. The homogenization effects diminish by assuming a polarised group of experts but are boosted by fully automating BAITs recommendations or considering one outsider in the group of experts.

6.2 Contributions

This paper makes five main contributions. The first three are direct contributions to the field of Decision Support Systems (DSS). The other two are methodological contributions. All contributions are theoretical in nature, but can also be adopted as practical contributions to end-users of BAIT.

6.2.1 Contributions to the field of DSS

The first contribution is the proof of usability of Bayesian inference in the field of DSS. Despite Bayesian inference being a common theme in the statistical literature, little was known on its application to DSS prior to this research. The particular interest in designing a Bayesian version of BAIT emerged from the curiosity of whether this method of statistical inference would be suitable for the newly introduced group of DSS called BAIT. Based on the findings of this paper, we believe that Bayesian inference is a more than appropriate alternative to conventional estimation routines for BAIT and other kinds of DSS. The enthusiasm about the potential of Bayesian methods in DSS originates from the attractive model assumptions of Bayesian inference and is amplified by the suitability for sequential updating and the competitive predictive performance and calibration whilst preserving a fast computation time. Therefore, we view this paper as a possible starting point for a larger adoption of Bayesian methods in DSS. By having designed a Bayesian methodology specifically for BAIT, we hope to have contributed to the further development of BAIT and hence also to its aspiration of becoming a breakthrough technology in the field of DSS.

Secondly, this research has contributed by being the first one to study the topic of feedback loops in the field of dynamic expert DSS. Here, we have taken BAIT as a primary example of dynamic expert DSS. Reviewing the literature in related fields enabled us to identify recommendation amplification and homogenization of expert behaviour as two pressing effects of the feedback loop in dynamic BAIT. Our simulations indicate that both effects are likely to occur due to applying a dynamic version of the technology. Therefore, we are of the opinion that these consequences should be more specifically addressed in future studies on dynamic expert DSS and should be accounted for by future end-users of these kinds of DSS and BAIT in particular.

The third contribution this research makes is exploring which conditions have a large influence on BAITs feedback loop effects. Our simulations provide evidence that the composition of the group of decision-makers is an influential factor on feedback loop effects. When BAIT is used for making a recommendation for a decision task based on a very heterogeneous or polarised group of decision-makers, one could find that these recommendations become rather conservative. Moreover, if the purpose of applying BAIT is to homogenize decisions of a very heterogeneous or polarized group, this purpose might not be fulfilled. End-users of BAIT should also account for too small groups of decision-makers, as this might lead to fluctuating recommendations. Furthermore, automating recommendations instead of supporting decisions using BAIT leads to very different effects. In case of full automation of recommendations, BAIT stops learning and completely homogenizes decisions.

In case of automating BAIT from a threshold value, recommendations might escalate rapidly and homogenization may be more moderate.

6.2.2 Methodological contributions

The first methodological contribution of this research is our self-designed, and hence unique, methodology for determining an informative prior for a binary decision task. The methodology uses the expected marginal effect of variables, base probability of the decision task and the expected confidence intervals of these expectations to obtain hyperparameters of a prior Gaussian density. Hereby, it offers a simple and effective way to transform prior beliefs into prior hyperparameters. A caveat of the methodology is its volatility with regard to the choice of input parameters. Nonetheless, this does not put a question mark against the methodology itself.

The last contribution this study makes is the methodology used for simulating the feedback loop of BAIT. Although several comparable studies were consulted in the design process, the majority of the methodology was self-designed for the particular use of BAIT. Despite that a plethora of different methods and assumptions could have been used, our methodology provides a proper example of a simulation framework. Especially the formulation of the agreement rate as a heterogeneity metric and using this metric to first simulate experts and later track the evolution of expert homogeneity may prove to be useful in future research. The simulation framework can be modified for the purpose of studying the feedback loop effects of different DSS. Besides, end-users of BAIT can also use the framework to explore the consequences of applying BAIT for their respective decision task.

6.3 Discussion and recommendations for future research

There are four topics for discussion. For each of these topics, this section provides the limitations and recommendations for future research.

6.3.1 Formalizing the informative prior

The prevailing limitation in the design of a Bayesian approach to BAIT lies in formalizing the prior distribution. Due to the many challenges faced in the design phase, no elaborate study has been conducted on the optimal design of a prior. This has led to approaches based on this informative prior being outperformed by MLE-based Bayesian approaches. Hence, we review the limitations of our informative prior design and use these to provide suggestions for the future.

Firstly, we have assumed a Gaussian density as the prior distribution. This choice was made for simplicity reasons, as Gaussian densities are widely known and offered a conjugate prior to Gibbs sampling for the probit model. Nevertheless, other types of prior densities could have been considered. Future studies could for instance consider the lognormal or truncated Gaussian densities, which allow for parameter values to be non-negative or non-positive. These densities could be useful, since experts are a priori already rather certain of the sign of the impact of variables.

Secondly, quite stringent assumptions were made on the input values for determining the informative prior hyperparameters. Here we explicitly refer to the choice of the expected marginal effects based on the prior importance, the base probability of a decision task and their respective

confidence intervals. In hindsight, these informative prior inputs were chosen too conservative. For future applications, we have provided insights into what would be more reasonable input values.

Thirdly, we have straightforwardly assumed an informative prior over a diffuse prior. This choice was made because of the small amount of data and the belief that no information should be left unused in a true Bayesian analysis. In retrospect, our informative prior inputs were chosen poorly, and according to Lee & Song (2004) in that case, it would have been better to use non-informative prior inputs. Later studies could thus investigate whether the model performance would be enhanced by applying a diffuse prior.

For constructing a prior density in future studies, we recommend studying a method called "elicitation" (O'Hagan et al., 2006). The elicitation of experts' knowledge can be used to construct a probability distribution that properly represents the expert's knowledge and uncertainty. Elicitation is appropriate to BAIT, as it is predominantly used in situations where prior information is appreciable and the data is limited. A possible approach is to first quantify each expert's opinion based on the decisions they have made in the experimental data set. Hereafter, the prior density can be derived as a function of all individual experts' distributions using various pooling techniques.

6.3.2 Methodology of Bayesian BAIT

The methodology used for Bayesian BAIT also consists of several limitations. As a starter, this research only focused on a binary decision task. Nevertheless, end-users have already requested the application of BAIT for multinomial or continuous decisions tasks. In order for Bayesian BAIT to be applied in a wider variety of settings, the application on other data structures should be explored.

Secondly, we have only considered standard logit and probit approaches as binary behavioural models. Other binary models could be explored, such as models that offer the possibility to include observed heterogeneity. These models would allow to segment the expert population based on certain characteristics, such as their experience level. These differences could be implemented by introducing expert-specific parameters into the model. Here, one could think of expert-specific intercepts or scale heterogeneity, which means including expert-specific variances of the residual term of the utility function.

Thirdly, only one method for sequential updating has been considered in our Bayesian approach. In this method, the posterior is approximated by a Gaussian distribution at every iteration and the posterior of the previous iteration is taken as the prior of the next iteration. This method introduces the risk of an accumulation of approximation errors. Although this has not yet led to large errors in our small sample environment, future researchers might be interested in finding more exact sequential Bayesian techniques. An example of such a technique is called *particle filtering*. This is a sequential Monte Carlo technique, in which the posteriors are not represented by a density function, but by "a cloud of particles", i.e. a set of random variables (Candy, 2016). Future research on Bayesian BAIT could investigate the applicability and performance of such a technique.

6.3.3 Methodology of the feedback loop simulations

The primary recommendation for a future study on BAITs feedback loop effects is to conduct a study in a real-life setting. Because this was not within the scope of this research, we had develop our own simulation framework. This framework became rather restricted by making many assumptions. In

this section, we address several topics on how the simulation scheme can be modified or improved.

Firstly, we have applied *MLE-based sequential VLR* as a dynamic version of BAIT, because of the conclusions drawn in the Bayesian BAIT chapter. Alternatively, we could have used fully Bayesian approaches, such as *batch - sequential VLR*. Due to the conservative and informative choice of our prior, we believe this would have led to smaller effects of individual observations. Therefore, the effects of the feedback loop would have become less apparent. Another alternative would be to apply one of the MCMC methods instead of VLR. Since the performance of these methods is very comparable, we do not expect any difference in the effects of the feedback loop. However, their longer computation times would have complicated the design phase of the methodology.

Secondly, the feedback loop was simplified to a version without advising experts, objective feedback or dynamic expert beliefs. Moreover, the decision process of experts was simplified to a weighted average between BAITs recommendation and expert decision probabilities. These assumptions are quite stringent, as experts are likely to have a more complex decision-making process that changes over time. They might receive training or objective feedback from which their knowledge base is updated. Moreover, the expert might become either more compliant to BAIT or more disobedient. Therefore, a relevant topic to study in the future is how experts alter their beliefs over time and how this impacts the feedback loop effects. In particular, one could study the incorporation of a Bayesian decision-making process of experts, as described by Chorus et al. (2009).

Thirdly, our simulated experts have been perceived as completely rational individuals. In our current simulations, no randomness was included in the expert decision behaviour. Moreover, all simulated experts were assigned an identical number of cases, implying an identical seniority and work ethic of all experts. Succeeding studies could explore the feedback loop by including randomness in the behaviour of experts and taking into account a varying seniority and work ethic.

6.3.4 Investigated feedback loop effects

The final limitation of our study concerns the investigated feedback loop effects. This study has only considered two possible effects. Future researches could explore other feedback loop effects, such as the evolution of expert dependency on BAIT.

A shortcoming of the investigation of BAITs recommendation amplification is that only five test cases have been considered. This number is limited, which makes drawing general conclusions risky. Furthermore, we have disregarded decision tasks with initial recommendations below the threshold of 0.5, since we assumed these would evolve in the opposite direction. Therefore, future research could review more test cases and also investigate cases with initial recommendations below this threshold. Another alternative would be to review the evolution of parameter values instead of recommendations for test cases.

Lastly, different metrics for tracking the homogenization effects of BAIT could have been considered. In this paper, the homogenization criterion was defined as the agreement rate of the two most heterogeneous experts. Other metrics could have been the average agreement rate of experts or a metric based on the final decision probability of experts. Moreover, in this research, we have refrained from the debate whether expert homogenization is a favourable effect. Homogenization is desired for many cases, since many end-users would apply BAIT to harmonize expert decisions. Nevertheless, homogenization becomes less favourable from the perspective that it might lead to less discussion and consultation between experts. Because this debate cannot straightforwardly be

settled, it would be wise to address it in a future study.

6.4 Recommendations to Councyl

This study has been commissioned by Councyl, the company that developed and commercialized BAIT. To further develop BAIT, Councyl was interested in developing a Bayesian approach to their current frequentist estimation routines. Moreover, the company wanted to learn more about the effects of the feedback loop caused by sequentially updating BAIT with real-life decisions. Hence, we will summarize the main recommendations for Councyl based on this study.

There are five assessment criteria upon which Councyl should assess the methodology of BAIT: (i) matching model assumptions; (ii) suitability for sequential updating; (iii) good predictive performance; (iv) good calibration of predicted probabilities; (v) short computation time. This study shows that some Bayesian approaches are more suitable for BAIT than the current MLE approach. This is predominantly based on the model assumptions and the suitability for sequential updating. In the short term, we recommend Councyl to incorporate the Bayesian approach called *MLE-based sequential Variational Logistic Regression (VLR)* to be able to sequentially update BAIT with real-life decisions. In the long term, we advise reviewing the methodology for determining an informative prior based on the prior inputs of experts. The Discussion section provides several ideas on how this specification can be improved. When the informative prior has been tuned correctly, we advise implementing the Bayesian approach called *Batch - sequential VLR*.

Based on our preliminary insights on the feedback loop, we make several recommendations. Firstly, BAITs recommendations are likely to be amplified over time. We advise Councyl to incorporate a "red flag" system into the technology, where Councyl is alerted in case of an extreme recommendation amplification. This could for instance be done by including several test cases in the technology, which recommendations are computed every time a new real-life decision is added to the model. When the recommendation deviates too much from the initial recommendation, by for instance $x\%$, Councyl should be alerted. The company could intervene by recalibrating the model, which could be done by discarding certain decisions or setting out new choice experiments. Secondly, our simulations indicate that homogenization of expert behaviour is likely to occur over time. The desirability of this has not been investigated in this research and should be decided upon by Councyl.

On a final note, we want to alert Councyl for some possible consequences of modelling conditions of BAIT. Firstly, BAIT might fail to initially provide extreme recommendations in case of a heterogeneous or polarized group of experts. Secondly, automating decisions from a threshold value will likely result in an extreme amplification of BAITs recommendations. Thirdly, fully automating BAITs decisions results in an extreme homogenization of expert behaviour. Fourthly, a small group of experts might lead to inconsistent and fluctuating recommendations. Lastly, BAIT will likely be able to harmonize the decisions of a group of experts with one outsider with opposing beliefs. However, this harmonization seems unlikely when there are two equally sized groups of experts with opposing beliefs.

References

Abdollahpouri, H. & Mansoury, M. (2020). Multi-sided exposure bias in recommendation. *arXiv preprint arXiv:2006.15772*.

Albers, C. J., Kiers, H. A., van Ravenzwaaij, D. & Savalei, V. (2018). Credible confidence: A pragmatic view on the frequentist vs bayesian debate. *Collabra: Psychology*, *4*(1).

Albert, J. H. & Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American statistical Association*, *88*(422), 669–679.

Amemiya, T. (1981). Qualitative response models: A survey. *Journal of economic literature*, *19*(4), 1483–1536.

Armstrong, J. S. (2001). *Principles of forecasting: a handbook for researchers and practitioners* (Vol. 30). Springer.

Bernardo, J. M. (1979). Reference posterior distributions for bayesian inference. *Journal of the Royal Statistical Society: Series B (Methodological)*, *41*(2), 113–128.

Bernardo, J. M. & Smith, A. F. (2009). *Bayesian theory* (Vol. 405). John Wiley & Sons.

Bishop, C. M. (2006). Pattern recognition. *Machine learning*, *128*(9).

Candy, J. V. (2016). *Bayesian signal processing: classical, modern, and particle filtering methods* (Vol. 54). John Wiley & Sons.

Caruana, R. & Niculescu-Mizil, A. (2004). Data mining in metric space: an empirical analysis of supervised learning performance criteria. In *Proceedings of the tenth acm sigkdd international conference on knowledge discovery and data mining* (pp. 69–78).

Chaney, A. J., Stewart, B. M. & Engelhardt, B. E. (2018). How algorithmic confounding in recommendation systems increases homogeneity and decreases utility. In *Proceedings of the 12th acm conference on recommender systems* (pp. 224–232).

Chib, S. & Carlin, B. P. (1999). On mcmc sampling in hierarchical longitudinal models. *Statistics and Computing*, *9*(1), 17–26.

Chib, S. & Greenberg, E. (1995). Understanding the metropolis-hastings algorithm. *The american statistician*, *49*(4), 327–335.

Chib, S. & Greenberg, E. (1996). Markov chain monte carlo simulation methods in econometrics. *Econometric theory*, *12*(3), 409–431.

Chicco, D. & Jurman, G. (2020). The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, *21*(1), 1–13.

Chorus, C. G., Arentze, T. A. & Timmermans, H. J. (2009). Traveler compliance with advice: A bayesian utilitarian perspective. *Transportation Research Part E: Logistics and Transportation Review*, *45*(3), 486–500.

Cook, S. R., Gelman, A. & Rubin, D. B. (2006). Validation of software for bayesian models using posterior quantiles. *Journal of Computational and Graphical Statistics*, *15*(3), 675–692.

D'Amour, A., Srinivasan, H., Atwood, J., Baljekar, P., Sculley, D. & Halpern, Y. (2020). Fairness is not static: deeper understanding of long term fairness via simulation studies. In *Proceedings of the 2020 conference on fairness, accountability, and transparency* (pp. 525–534).

Daziano, R. A. & Bolduc, D. (2013). Covariance, identification, and finite-sample performance of the msl and bayes estimators of a logit model with latent attributes. *Transportation*, *40*(3), 647–670.

De Metz, J., Thoral, P. J., Chorus, C. G., Elbers, P. W. & van den Bogaard, B. (2021). Behavioural artificial intelligence technology for covid-19 intensivist triage decisions: making the implicit explicit. *Intensive care medicine*, 1–2.

Doron, J. & Gaudreau, P. (2014). A point-by-point analysis of performance in a fencing match: Psychological processes associated with winning and losing streaks. *Journal of Sport and Exercise Psychology*, *36*(1), 3–13.

Edwards, W. (1954). The theory of decision making. *Psychological bulletin*, *51*(4), 380.

Ensign, D., Friedler, S. A., Neville, S., Scheidegger, C. & Venkatasubramanian, S. (2018). Runaway feedback loops in predictive policing. In *Conference on fairness, accountability and transparency* (pp. 160–171).

Frühwirth-Schnatter, S. & Frühwirth, R. (2010). Data augmentation and mcmc for binary and multinomial logit models. In *Statistical modelling and regression structures* (pp. 111–132). Springer.

Fussl, A., Fruehwirth-Schnatter, S. & Fruehwirth, R. (2013). Efficient mcmc for binomial logit models. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, *23*(1), 1–21.

Gelfand, A. E. & Smith, A. F. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, *85*(410), 398–409.

Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models (comment on article by browne and draper). *Bayesian analysis*, *1*(3), 515–534.

Gelman, A. (2008). Scaling regression inputs by dividing by two standard deviations. *Statistics in medicine*, *27*(15), 2865–2873.

Gelman, A., Carlin, J. B., Stern, H. S. & Rubin, D. B. (1995). *Bayesian data analysis*. Chapman and Hall/CRC.

Gelman, A., Gilks, W. R. & Roberts, G. O. (1997). Weak convergence and optimal scaling of random walk metropolis algorithms. *The annals of applied probability*, *7*(1), 110–120.

Gelman, A., Simpson, D. & Betancourt, M. (2017). The prior can often only be understood in the context of the likelihood. *Entropy*, *19*(10), 555.

Geman, S. & Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*(6), 721–741.

Greenberg, E. (2012). *Introduction to bayesian econometrics*. Cambridge University Press.

Gretton, C. (2018). Trust and transparency in machine learning-based clinical decision support. In *Human and machine learning* (pp. 279–292). Springer.

Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications.

Hausman, J. A. & Wise, D. A. (1978). A conditional probit model for qualitative choice: Discrete decisions recognizing interdependence and heterogeneous preferences. *Econometrica: Journal of the econometric society*, 403–426.

Held, L. & Holmes, C. C. (2006). Bayesian auxiliary variable models for binary and multinomial regression. *Bayesian analysis*, *1*(1), 145–168.

Jaakkola, T. S. & Jordan, M. I. (2000). Bayesian parameter estimation via variational methods. *Statistics and Computing*, *10*(1), 25–37.

Jackman, S. (2009). *Bayesian analysis for the social sciences* (Vol. 846). John Wiley & Sons.

Jiang, R., Chiappa, S., Lattimore, T., György, A. & Kohli, P. (2019). Degenerate feedback loops in recommender systems. In *Proceedings of the 2019 aaai/acm conference on ai, ethics, and society* (pp. 383–390).

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S. & Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, *37*(2), 183–233.

Kaplan, D. & Depaoli, S. (2013). Bayesian statistical methods. *Oxford handbook of quantitative methods*, 407–437.

Kass, R. E. & Wasserman, L. (1996). The selection of prior distributions by formal rules. *Journal of the American statistical Association*, *91*(435), 1343–1370.

Kiesling, E., Günther, M., Stummer, C. & Wakolbinger, L. M. (2012). Agent-based simulation of innovation diffusion: a review. *Central European Journal of Operations Research*, *20*(2), 183–230.

Kim, Y.-J., Ahn, K.-U. & Park, C.-S. (2014). Decision making of hvac system using bayesian markov chain monte carlo method. *Energy and Buildings*, *72*, 112–121.

Kliem, S., Kröger, C. & Kosfelder, J. (2010). Dialectical behavior therapy for borderline personality disorder: a meta-analysis using mixed-effects modeling. *Journal of consulting and clinical psychology*, *78*(6), 936.

Kuhn, M. & Johnson, K. (2013). Measuring performance in classification models. In *Applied predictive modeling* (pp. 247–273). Springer.

Larrick, R. P. & Feiler, D. C. (2015). Expertise in decision making. *The Wiley Blackwell handbook of judgment and decision making*, *2*, 696–721.

Lee, S.-Y. & Song, X.-Y. (2004). Evaluation of the bayesian and maximum likelihood approaches in analyzing structural equation models with small sample sizes. *Multivariate Behavioral Research*, *39*(4), 653–686.

Lum, K. & Isaac, W. (2016). To predict and serve? *Significance*, *13*(5), 14–19.

MacNab, Y. C., Malloy, D. C., Hadjistavropoulos, T., Sevigny, P. R., McCarthy, E. F., Murakami, M., ... Liu, P. L. (2011). Idealism and relativism across cultures: A cross-cultural examination of physicians' responses on the ethics position questionnaire (epq). *Journal of Cross-Cultural Psychology*, *42*(7), 1272–1278.

Mansoury, M., Abdollahpouri, H., Pechenizkiy, M., Mobasher, B. & Burke, R. (2020). Feedback loop and bias amplification in recommender systems. In *Proceedings of the 29th acm international conference on information & knowledge management* (pp. 2145–2148).

Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, *405*(2), 442–451.

McFadden, D. & Train, K. (2000). Mixed mnl models for discrete response. *Journal of applied Econometrics*, *15*(5), 447–470.

McNeish, D. (2016). On using bayesian methods to address small sample problems. *Structural Equation Modeling: A Multidisciplinary Journal*, *23*(5), 750–773.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. & Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, *21*(6), 1087–1092.

Montani, S. & Striani, M. (2019). Artificial intelligence in clinical decision support: a focused literature survey. *Yearbook of medical informatics*, *28*(01), 120–127.

Nguyen, T. T., Hui, P.-M., Harper, F. M., Terveen, L. & Konstan, J. A. (2014). Exploring the filter bubble: the effect of using recommender systems on content diversity. In *Proceedings of the 23rd international conference on world wide web* (pp. 677–686).

O'Hagan, A., Buck, C. E., Daneshkhah, A., Eiser, J. R., Garthwaite, P. H., Jenkinson, D. J., ... Rakow, T. (2006). Uncertain judgements: eliciting experts' probabilities.

Raftery, A. E. (1996). Approximate bayes factors and accounting for model uncertainty in generalised linear models. *Biometrika*, *83*(2), 251–266.

Raftery, A. E. & Lewis, S. M. (1995). The number of iterations, convergence diagnostics and generic metropolis algorithms. *Practical Markov Chain Monte Carlo*, *7*(98), 763–773.

Robert, C. P., Elvira, V., Tawn, N. & Wu, C. (2018). Accelerating mcmc algorithms. *Wiley Interdisciplinary Reviews: Computational Statistics*, *10*(5), e1435.

Robert, C. P. et al. (2007). *The bayesian choice: from decision-theoretic foundations to computational implementation* (Vol. 2). Springer.

Saito, T. & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, *10*(3), e0118432.

Schrama, V. (2021). Transparent decision support in ever-changing healthcare contexts: Designing an architecture of a transparent and dynamic clinical decision support system grounded in discrete choice modeling.

Spiegelhalter, D. J., Dawid, A. P., Lauritzen, S. L. & Cowell, R. G. (1993). Bayesian analysis in expert systems. *Statistical science*, 219–247.

Stenling, A., Ivarsson, A., Johnson, U. & Lindwall, M. (2015). Bayesian structural equation modeling in sport and exercise psychology. *Journal of Sport and Exercise Psychology*, *37*(4), 410–420.

Tang, T., Ghorbani, A., Squazzoni, F. & Chorus, C. G. (2021). Together alone: a group-based polarization measurement. *Quality & Quantity*, 1–33.

Tanner, M. A. (2012). *Tools for statistical inference*. Springer.

Tanner, M. A. & Wong, W. H. (1987). The calculation of posterior distributions by data augmentation. *Journal of the American statistical Association*, *82*(398), 528–540.

Ten Broeke, A., Hulscher, J., Heyning, N., Kooi, E. & Chorus, C. (2021). Bait: A new medical decision support technology based on discrete choice theory. *Medical Decision Making*, 0272989X211001320.

Tierney, L. (1994). Markov chains for exploring posterior distributions. *the Annals of Statistics*, 1701–1728.

Train, K. E. (2009). *Discrete choice methods with simulation*. Cambridge university press.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., ... Altman, R. B. (2001). Missing value estimation methods for dna microarrays. *Bioinformatics*, *17*(6), 520–525.

Walsh, A. (1987). Teaching understanding and interpretation of logit regression. *Teaching sociology*, 178–183.

Wanless, S. B., Rimm-Kaufman, S. E., Abry, T., Larsen, R. A. & Patton, C. L. (2015). Engagement in training as a mechanism to understanding fidelity of implementation of the responsive classroom approach. *Prevention Science*, *16*(8), 1107–1116.

Yildiz, C. (2021). The human in command: An exploratory study into human moral autonomy of behavioural artificial intelligence technology.

Zhang, S. (2012). Nearest neighbor selection for iteratively knn imputation. *Journal of Systems and Software*, *85*(11), 2541–2552.

Appendix A

Overview of algorithms

This chapter provides an overview of all algorithms used in this paper. Sections A.1 to A.4 are involved with algorithms used for the formulation of our Bayesian approaches in Section 4.4. Section A.5 describes the methodology for simulating data matrices. Finally, Section provides the algorithm for simulating expert beliefs under varying conditions.

A.1 Formalizing the prior distribution

Algorithm 1 is an overview of the methodology that we derived in Section 4.4.4 to specify prior parameters.

Algorithm 1 Specification of prior parameters

input EMEs and LBEMEs of independent variables, P_{base}, LBP_{base}
$U_{base} = H^{-1}(P_{base})$
for $j \in [1:p]$ **do**
  if vartype(j) is ordinal with N_{ord} linear levels with interpolation **then**
    $\tilde{\sigma}_j = \frac{1}{1.96}\left(\tilde{\beta}_j - \frac{\text{LBEME}_j}{h(U_{base})}\right)$
    $\tilde{\beta}_j = \frac{\text{EME}_j}{h(U_{base})}$
    $\tilde{x}_j = \frac{1}{N_{ord}}$
  else
    $\tilde{\beta}_j = H^{-1}\left(\text{EME}_j + H(U_{base})\right) - U_{base}$
    $\tilde{\sigma}_j = \frac{1}{1.96}\left(U_{base} + \tilde{\beta}_j - H^{-1}\left(LBEME_j + H(U_{base})\right)\right)$
    $\tilde{x}_j = \frac{a+b}{2}$ where $[a, b]$ is the interval on which x_j is defined
  end if
end for
$\tilde{\beta}_0 = U_{base} - \tilde{\boldsymbol{x}}'_{1:p}\tilde{\boldsymbol{\beta}}_{1:p}$
$\tilde{\sigma}_0 = \frac{1}{1.96}\left(U_{base} - H^{-1}(LBP_{base})\right)$
Set covariance matrix $\tilde{\Sigma}$ as a diagonal matrix with elements $\tilde{\sigma}_0, \tilde{\sigma}_1, \ldots, \tilde{\sigma}_p$
return $\tilde{\boldsymbol{\beta}}, \tilde{\Sigma}$

A.2 Gibbs sampler for the probit model

Algorithm 2 provides the sampling scheme derived in Section 4.4.5 to yield posterior parameters $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\Sigma}}$ for the probit model using the Gibbs sampler.

Algorithm 2 Probit Gibbs sampling scheme

input $\boldsymbol{X}$, $\boldsymbol{y}$, $\tilde{\boldsymbol{\beta}}^P$, $\tilde{\boldsymbol{B}}^P$, $N_{burn}, n_{thin}, N_{sim}$
$N_{total} = N_{burn} + N_{sim} \cdot n_{thin}$
$\boldsymbol{\beta}^{(0)} = \tilde{\boldsymbol{\beta}}^P$
$\boldsymbol{B}^{(0)} = \tilde{\boldsymbol{B}}^P$
$\hat{\boldsymbol{B}} = \left((\boldsymbol{B}^{(0)})^{-1} + \boldsymbol{X}'\boldsymbol{X}\right)^{-1}$
Ω = empty set of stored $\boldsymbol{\beta}$'s
for $m \in [1 : N_{total}]$ **do**
  for $i \in [1 : N]$ **do**
    Draw $z_i^{(m)}$ from $\begin{cases} N(\boldsymbol{x}_i'\boldsymbol{\beta}^{(m-1)}, 1) \text{ truncated at the left by } 0 & \text{if } y_i = 1 \\ N(\boldsymbol{x}_i'\boldsymbol{\beta}^{(m-1)}, 1) \text{ truncated at the right by } 0 & \text{if } y_i = 0 \end{cases}$
  end for
  $\boldsymbol{b}^{(m)} = \hat{\boldsymbol{B}}\left((\boldsymbol{B}^{(0)})^{-1}\boldsymbol{\beta}^{(0)} + \boldsymbol{X}'\boldsymbol{z}\right)$
  Draw $\boldsymbol{\beta}^{(m)}$ from $N(\boldsymbol{b}^{(m)}, \hat{\boldsymbol{B}})$
  if $m > N_{burn}$ & $m \,\%\, n_{thin} = 0$ **then** add $\boldsymbol{\beta}^{(m)}$ to Ω
  end if
end for
$\hat{\boldsymbol{\beta}}$ = mean of all $\boldsymbol{\beta}$'s in Ω
$\hat{\boldsymbol{\Sigma}}$ = covariance matrix of all $\boldsymbol{\beta}$'s in Ω
return $\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\Sigma}}$

A.3 Metroplis-Hastings sampler for the logit model

Algorithm 3 provides the sampling scheme derived in Section 4.4.6 to yield posterior parameters $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\Sigma}}$ for the logit model using the Metropolis-Hastings sampler.

Algorithm 3 Metropolis-Hastings sampling scheme

input $\boldsymbol{X}$, $\boldsymbol{y}$, $\tilde{\boldsymbol{\beta}}^L$, $\tilde{\boldsymbol{B}}^L$, $N_{burn}, n_{thin}, N_{sim}$
$N_{total} = N_{burn} + N_{sim} \cdot n_{thin}$
$\boldsymbol{b}^{(0)} = \tilde{\boldsymbol{\beta}}^L$
$\boldsymbol{B}^{(0)} = \tilde{\boldsymbol{B}}^L$
$\boldsymbol{\beta}^{(1)} = \tilde{\boldsymbol{\beta}}^L$
$\hat{\boldsymbol{B}} = \left((\boldsymbol{B}^{(0)})^{-1} + \frac{3}{\pi^2}\boldsymbol{X}'\boldsymbol{X}\right)^{-1}$
Ω = empty set of stored $\boldsymbol{\beta}$'s
for $m \in [1 : N_{total}]$ **do**
  for $i \in [1 : N]$ **do**

$$z_i|\boldsymbol{\beta}, y_i \sim \begin{cases} G(\boldsymbol{x}_i'\boldsymbol{\beta})\mathbf{1}\{z_i > 0\} & \text{if } y_i = 1 \\ G(\boldsymbol{x}_i'\boldsymbol{\beta})\mathbf{1}\{z_i \leq 0\} & \text{if } y_i = 0 \end{cases}$$

  end for
  Set $\boldsymbol{b}^{(m)} = \hat{\boldsymbol{B}}\left((\boldsymbol{B}^{(0)})^{-1}\boldsymbol{b}^{(0)} + \frac{3}{\pi^2}\boldsymbol{X}'\boldsymbol{z}\right)$
  Draw candidate $\boldsymbol{\beta}^*$ from proposal density $q(\boldsymbol{\beta}^*|\boldsymbol{z}) = N(\boldsymbol{b}^{(m)}, \hat{\boldsymbol{B}})$
  Determine likelihood of $p(\boldsymbol{z}|\boldsymbol{\beta}^*)$ and $p(\boldsymbol{z}|\boldsymbol{\beta}^{(m)})$ by $p(\boldsymbol{z}|\boldsymbol{\beta}) = \prod_{i=1}^{N} g(z_i - \boldsymbol{x}_i'\boldsymbol{\beta})$
  Determine probability of move $\alpha(\boldsymbol{\beta}^*|\boldsymbol{\beta}^{(m)}, \boldsymbol{z}) = \min\left[\frac{p(\boldsymbol{z}|\boldsymbol{\beta}^*)p(\boldsymbol{\beta}^*)q(\boldsymbol{\beta}^{(m)}|\boldsymbol{z})}{p(\boldsymbol{z}|\boldsymbol{\beta}^{(m)})p(\boldsymbol{\beta}^{(m)})q(\boldsymbol{\beta}^*|\boldsymbol{z})}, 1\right]$
  Draw u from $Uniform[0, 1]$

$$\text{Set } \boldsymbol{\beta}^{(m+1)} = \begin{cases} \boldsymbol{\beta}^* & \text{if } u \leq \alpha(\boldsymbol{\beta}^*|\boldsymbol{\beta}^{(m)}, \boldsymbol{z}) \\ \boldsymbol{\beta}^{(m)} & \text{otherwise} \end{cases}$$

  if $m > N_{burn}$ & $m \,\%\, n_{thin} = 0$ **then** add $\boldsymbol{\beta}^{(m+1)}$ to Ω
  end if
end for
$\hat{\boldsymbol{\beta}}$ = mean of all $\boldsymbol{\beta}$'s in Ω
$\hat{\boldsymbol{\Sigma}}$ = covariance matrix of all $\boldsymbol{\beta}$'s in Ω
return $\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\Sigma}}$

A.4 Variational Logistic Regression

Algorithm 4 provides the sampling scheme derived in Section 4.4.7 to yield posterior parameters $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\Sigma}}$ for the logit model using batch Variational Logistic Regression. Algorithm 5 provides a similar sampling scheme, but for sequential Variational Logistic Regression.

Algorithm 4 Batch Variational Logistic Regression scheme

input $\boldsymbol{X}$, $\boldsymbol{y}$, $\tilde{\boldsymbol{\beta}}^L$, $\tilde{\boldsymbol{B}}^L$
$\hat{\boldsymbol{\beta}}^{new} = \tilde{\boldsymbol{\beta}}^L$
$\hat{\boldsymbol{B}} = \tilde{\boldsymbol{B}}^L$
while $(\hat{\boldsymbol{\beta}}^{new} - \hat{\boldsymbol{\beta}}^{old}) > 0.001$ **do**
  for $i \in [1:N]$ **do** $\xi_i = \sqrt{\boldsymbol{x}_i'(\hat{\boldsymbol{B}} + \hat{\boldsymbol{\beta}}^{new}\hat{\boldsymbol{\beta}}^{new\prime})\boldsymbol{x}_i}$
  end for
  $\hat{\boldsymbol{B}} = \left((\tilde{\boldsymbol{B}}^L)^{-1} + 2\sum_{i=1}^N \lambda(\xi_i)\boldsymbol{x}_i\boldsymbol{x}_i'\right)^{-1}$
  $\hat{\boldsymbol{\beta}}^{new} = \hat{\boldsymbol{B}}\left((\tilde{\boldsymbol{B}}^L)^{-1}\tilde{\boldsymbol{\beta}}^L) + \sum_{i=1}^N (y_i - \frac{1}{2})\boldsymbol{x}_i\right)$
  $\hat{\boldsymbol{\beta}}^{old} = \hat{\boldsymbol{\beta}}^{new}$
end while
return $\hat{\boldsymbol{\beta}}^{new}, \hat{\boldsymbol{B}}$

Algorithm 5 Sequential Variational Logistic Regression scheme

input $\boldsymbol{X}$, $\boldsymbol{y}$, $\tilde{\boldsymbol{\beta}}^L$, $\tilde{\boldsymbol{B}}^L$
$\hat{\boldsymbol{\beta}}^{(0)} = \tilde{\boldsymbol{\beta}}^L$
$\hat{\boldsymbol{B}}^{(0)} = \tilde{\boldsymbol{B}}^L$
for $i \in [1:N]$ **do**
  $\xi^{(i)} = \sqrt{\boldsymbol{x}_i'(\hat{\boldsymbol{B}}^{(i-1)} + \hat{\boldsymbol{\beta}}^{(i-1)}\hat{\boldsymbol{\beta}}^{(i-1)\prime})\boldsymbol{x}_i}$
  $\hat{\boldsymbol{B}}^{(i)} = \left(\left(\hat{\boldsymbol{B}}^{(i-1)}\right)^{-1} + 2\sum_{i=1}^N \lambda(\xi_i)\boldsymbol{x}_i\boldsymbol{x}_i'\right)^{-1}$
  $\hat{\boldsymbol{\beta}}^{(i)} = \hat{\boldsymbol{B}}^{(i)}\left(\left(\hat{\boldsymbol{B}}^{(i-1)}\right)^{-1}\hat{\boldsymbol{\beta}}^{(i-1)} + (y_i - \frac{1}{2})\boldsymbol{x}_i\right)$
end for
return $\hat{\boldsymbol{\beta}}^{(N)}, \hat{\boldsymbol{B}}^{(N)}$

A.5 Data simulation

Algorithm 6 provides the methodology for simulating new data matrices, which is used in Section 5.4.1

Algorithm 6 Simulation of new data matrix $X^{(sim)}$

input N_{sim} = size of simulated matrix, vartype = variable types of all input variables
$\boldsymbol{x}_0$ = vector of ones of length N_{sim}
$X = [\boldsymbol{x}_0]$
for $j \in [1 : p]$ **do**
  if vartype(j) is binary **then**
    $\boldsymbol{x}_j = N_{sim}$ random draws from $Uniform\{0, 1\}$
  else if vartype(j) is ordinal with N_{ord} linear levels **then**
    $\boldsymbol{x}_j = N_{sim}$ random draws from $\frac{1}{N_{ord}} Uniform\{0, N_{ord}\}$
  else if vartype(j) is ordinal with N_{ord} linear levels with interpolation **then**
    $\boldsymbol{x}_j = N_{sim}$ random draws from $\frac{1}{N_{ord}} Uniform[0, N_{ord}]$
  else if vartype(j) is ordinal with nonlinear levels **then**
    $\boldsymbol{z}_j = N_{sim}$ random draws from $Uniform\{0, N_{ord}\}$
    From $\boldsymbol{z}_j$ dummy encode $\boldsymbol{x}_2, \ldots, \boldsymbol{x}_{N_{ord}}$ with $\boldsymbol{x}_1$ as base level
  end if
  $X = [X \quad \boldsymbol{x}_j]$
end for
return X

A.6 Simulation of expert beliefs

Algorithm 7 states the methodology used in Section 5.4 to simulate expert belief vectors $\boldsymbol{\beta}_1^{expert}$, $\ldots$, $\boldsymbol{\beta}_{N_{simexp}}^{expert}$. These vectors are simulated according to scaling factor κ of Σ^{MLE}. This scaling factor is determined such that $\overline{AR}_0 \approx AR_{goal}$. Since there is no exact solution for κ, we restrict ourselves in the evaluation of κ to steps of size 0.1. For the regular experts, we set AR_{goal} equal to the $\overline{AR}_0$ of the actual experts, which is 0.749. Moreover, we set AR_{goal} to 0.9 for the homogeneous group and to 0.5 for the heterogeneous group.

Algorithm 7 Simulation of $\boldsymbol{\beta}_1^{expert}, \ldots, \boldsymbol{\beta}_{N_{simexp}}^{expert}$ according to scaling factor κ

input Σ^{MLE}, $\boldsymbol{\beta}^{MLE}$, N_{simexp}, AR_{goal} = goal of AR_0, X_0 = matrix of D_{hyp} hypothetical scenarios
$\kappa = 0$
$\overline{AR}_0 = \infty$
while $\overline{AR}_0 > AR_{goal}$ **do**
  $\kappa = \kappa + 0.1$
  for $i \in [1 : N_{simexp}]$ **do**
    $\boldsymbol{\beta}_1^{expert}, \ldots, \boldsymbol{\beta}_{N_{simexp}}^{expert} \sim N(\boldsymbol{\beta}^{MLE}, \kappa\Sigma^{MLE})$
    for $j \in [1 : D_{hyp}]$ **do**
      $P_{i,0,j}^{expert} = G(\boldsymbol{x}_{0,j}'\boldsymbol{\beta}_i^{expert})$
      $y_{i,0,j} = \begin{cases} 1 & \text{if } P_{i,0,j}^{expert} > T \\ 0 & \text{if } P_{i,0,j}^{expert} \leq T \end{cases}$
    end for
  end for
  $\overline{AR}_0 = \frac{1}{\frac{1}{2}N_{simexp}(N_{simexp}-1)} \frac{1}{D_{hyp}} \sum_{r=1}^{N_{simexp}-1} \sum_{s=r+1}^{N_{simexp}} \sum_{l=1}^{D_{hyp}} \mathbf{1}\{y_{r,0,l} = y_{s,0,l}\}$
end while
return $\kappa, \overline{AR}_0, \boldsymbol{\beta}_1^{expert}, \ldots, \boldsymbol{\beta}_{N_{simexp}}^{expert}$

Appendix B

Validation of Bayesian BAIT software

In a Bayesian statistical model, and in particular, in Markov Chain Monte Carlo (MCMC) methods, there is a large chance that errors in the software implementation occur. In order to ensure that the Bayesian BAIT model is correctly implemented, the model should be validated. Therefore, we have applied the simulation-based method for software validation of Bayesian methods as described by Cook et al. (2006). The validation procedure is built on the following idea: when we draw parameters from the prior distribution and draw data from their sampling distribution given the drawn parameters, and then perform Bayesian inference correctly, the resulting posterior inferences will on average be correct.

B.1 Methodology

Consider Bayesian model $p(y|\boldsymbol{\beta})p(\boldsymbol{\beta})$, where $p(y|\boldsymbol{\beta})$ represents the likelihood of the data y and $p(\boldsymbol{\beta})$ represents the prior distribution of parameter vector $\boldsymbol{\beta}$. We first simulate matrix matrix $X^{(sim)}$ according to Algorithm 6 provided in Appendix A.5.. We then apply N_{rep} replications of the following procedure.

1. We sample "true" parameter value $\boldsymbol{\beta}^{(0)}$ from prior distribution $p(\boldsymbol{\beta})$;

2. We sample vector y_{sim} according to data generating process $p(y|\boldsymbol{\beta})$. As y_i are binary, the values are generated from Bernoulli distribution $Bernoulli(p_i)$. For a logit model we have $p_i^L = \frac{exp(\boldsymbol{x}_i'\boldsymbol{\beta})}{1+exp(\boldsymbol{x}_i'\boldsymbol{\beta})}$, whereas under a probit model $p_i^P = \Phi(\boldsymbol{x}_i'\boldsymbol{\beta})$;

3. We apply the MCMC sampler, which gives us the posterior sample $\boldsymbol{\beta}^{(1)}, \ldots, \boldsymbol{\beta}^{(L)}$. If the software is written correctly, $\boldsymbol{\beta}^{(1)}, \ldots, \boldsymbol{\beta}^{(L)}$ are a sample from $p(\boldsymbol{\beta}|y)$;

4. We estimate empirical quantile of the kth replication by $\hat{q}_k(\beta_j^{(0)}) = \frac{1}{L}\sum_{l=1}^{L} \mathbf{1}\{\beta_j^{(0)} > \beta_j^{(l)}\}$.

The theorem of Cook et al. (2006) states that the software is written correctly when the distribution of $\hat{q}(\beta_j^{(0)})$ approaches $Uniform(0,1)$ for large values of L.

Moreover, this theorem suggests that when the software works properly, and therefore the empirical quantiles are uniformly distributed, $\Phi^{-1}(q)$ should follow a standard normal distribution. To test whether $\Phi^{-1}(q)$ follows a standard normal distribution, we introduce test statistic $\chi_j^2 = \sum_{i=1}^{N_{rep}} \left(\Phi^{-1}(q_i)\right)^2$. This test statistic should follow a χ^2-distribution with N_{rep} degrees of

freedom when the software works correctly. To test this, we quantify the posterior quantiles' deviation from uniformity by calculating the associated p-value p_j for each χ^2_j. Ultimately, the p_j's are transformed into $z_j = \Phi^{-1}(p_j)$, which absolute values are then plotted. When no z_j statistics are extreme, there is no indication of incorrectly written software.

B.2 Results

In this paper, we have applied four different MCMC samplers: the batch Gibbs sampler for the probit model, the sequential Gibbs sampler for the probit model, the batch Metropolis-Hastings sampler for the logit model and the sequential Metropolis-Hastings sampler for the logit model. We have validated these models for the COVID-19 case. This means that we have used the prior knowledge and distributions of the variables in the COVID-19 data set. Therefore, the number of variables p is equal to 16. Based on the properties of the variables in the COVID-19 data set, we used Algorithm 6 to simulate matrix $X^{(sim)}$. We have tested the MCMC samplers for two different sample sizes. First, we have validated the model on a data matrix of size $N_{sim} = 100$, which is in between the sample sizes of experimental and real-life data sets. Secondly, we also wanted to test whether the model performs well for very small sample sizes, which is why we have tested them for $N_{sim} = 5$.

To validate the MCMC samplers, we have set the number of replications of the procedure to $N_{rep} = 500$. For every replication, we then obtained a posterior sample of size $L = 1000$. As stated in the theorem of Cook et al. (2006), the software is written correctly when the distribution of $\hat{q}(\beta_j^{(0)})$ approaches $Uniform(0, 1)$.

B.2.1 Validation of the batch Gibbs sampler for the probit model

The first model we have tested is the batch Gibbs sampler for the probit model. For both sample sizes $N_{sim} = 5$ and $N_{sim} = 100$ we have found no indication of incorrectly written software. We have obtained histograms of all empirical quantiles $\hat{q}(\beta_0^{P,(0)}), \ldots, \hat{q}(\beta_p^{P,(0)})$. Exemples of these plots can be found in Figure B.3, which show the histograms of empirical quantiles $\hat{q}(\beta_1^{P,(0)})$ and $\hat{q}(\beta_2^{P,(0)})$ for $N_{sim} = 100$. In this figure we see that the empirical quantiles are approximately uniformly distributed. Furthermore, a plot of the absolute values of the $z_0^P, z_1^P, \ldots, z_p^P$ statistics can be found in Figure B.4. As no z_j^P statistics attain an extreme value, we have no reason to believe the software was written incorrectly.

Figure B.1: The posterior quantiles of β_1^P and β_2^P of the Gibbs sampler for the probit model closely resemble a uniform distribution

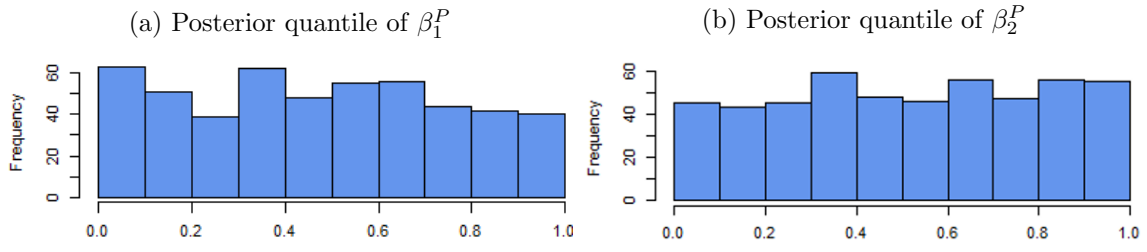
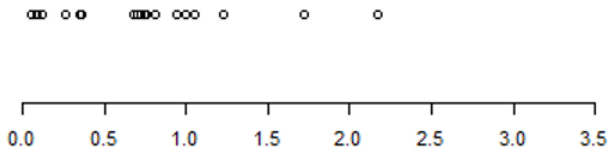


Figure B.2: The absolute z transformation of p_j^P values of the Gibbs sampler for the probit model do not attain any extreme values



B.2.2 Validation of the sequential Gibbs sampler for the probit model

Secondly, we have tested the sequential Gibbs sampler for the probit model for sample sizes $N_{sim} = 5$ and $N_{sim} = 100$. Figure B.3 displays an exploratory plot of empirical quantiles $\hat{q}(\beta_1^{P,(0)})$ and $\hat{q}(\beta_2^{P,(0)})$ for $N_{sim} = 5$. This figure does not provide any indication of the empirical quantiles not being uniformly distributed. A plot of the subsequent absolute values of the $z_0^P, z_1^P, \ldots, z_p^P$ statistics can be found in Figure B.4. No z_j^P statistics attain an extreme value, there is no indication of incorrectly written software.

Figure B.3: The posterior quantiles of β_1^P and β_2^P of the sequential Gibbs sampler for the probit model closely resemble a uniform distribution

(a) Posterior quantile of β_1^P (b) Posterior quantile of β_2^P

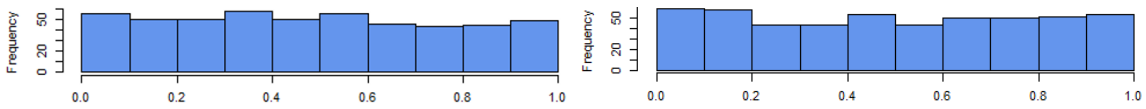
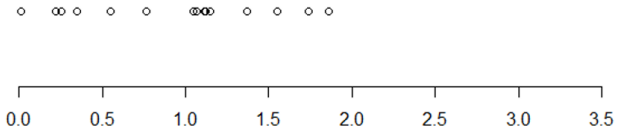


Figure B.4: The absolute z transformation of p_j^P values of the sequential Gibbs sampler for the probit model do not attain any extreme values

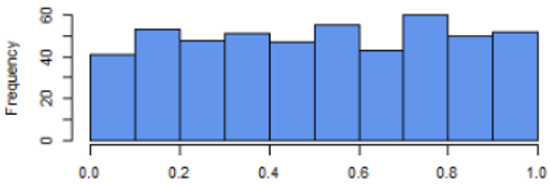


B.2.3 Validation of batch Metropolis-Hastings sampler for the logit model

Thirdly, we have tested the batch Metropolis-Hastings sampler for the logit model for both sample sizes. Figure B.7 shows an exploratory plot of empirical quantiles $\hat{q}(\beta_1^{L,(0)})$ and $\hat{q}(\beta_2^{L,(0)})$ for $N_{sim} = 100$. The figure indicates that the empirical quantiles are approximately uniformly distributed. Stronger evidence is provided in Figure B.4, which shows a plot of the absolute values of the $z_0^L, z_1^L, \ldots, z_p^L$ statistics. None of the z_j^L statistics attains an extreme value and hence there is no indication of incorrectly written software.

Figure B.5: The posterior quantiles of β_1^L and β_2^L of the batch Metropolis-Hastings sampler for the logit model closely resemble a uniform distribution

(a) Posterior quantile of β_1^L



(b) Posterior quantile of β_2^L

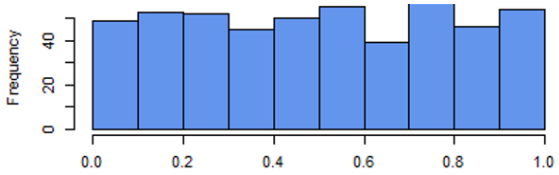
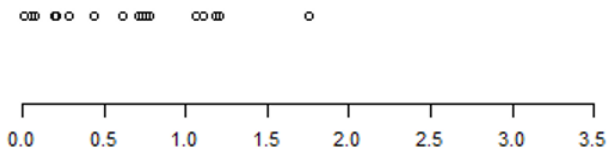


Figure B.6: The absolute z transformation of p_j^L values of the batch Metropolis-Hastings sampler for the logit model do not attain any extreme values

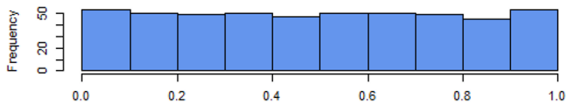


B.2.4 Validation of sequential Metropolis-Hastings sampler for the logit model

Lastly, we have tested the sequential Metropolis-Hastings sampler for the logit model for the small and regular sample size. In Figure B.7, which shows an exploratory plot of empirical quantiles $\hat{q}(\beta_1^{L,(0)})$ and $\hat{q}(\beta_2^{L,(0)})$ for $N_{sim} = 5$, we can find no indication of non-uniform distributions. Figure B.4 shows a plot of the absolute values of the $z_0^L, z_1^L, \ldots, z_p^L$ statistics. Although the most extreme z_j^L statistic is slightly larger than for the other models, it does not attain an extreme value. This means that we believe the sequential Metropolis-Hastings sampler works correctly, also for very small sample sizes.

Figure B.7: The posterior quantiles of β_1^L and β_2^L of the sequential Metropolis-Hastings sampler for the logit model closely resemble a uniform distribution

(a) Posterior quantile of β_1^L



(b) Posterior quantile of β_2^L

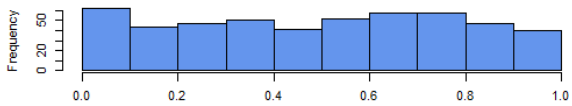
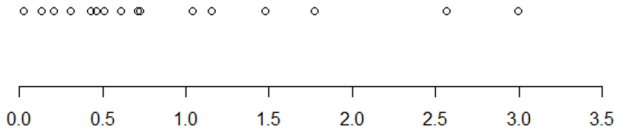


Figure B.8: The absolute z transformation of p_j^L values of the sequential Metropolis-Hastings sampler for the logit model do not attain any extreme values



Appendix C

Plots of results Bayesian BAIT

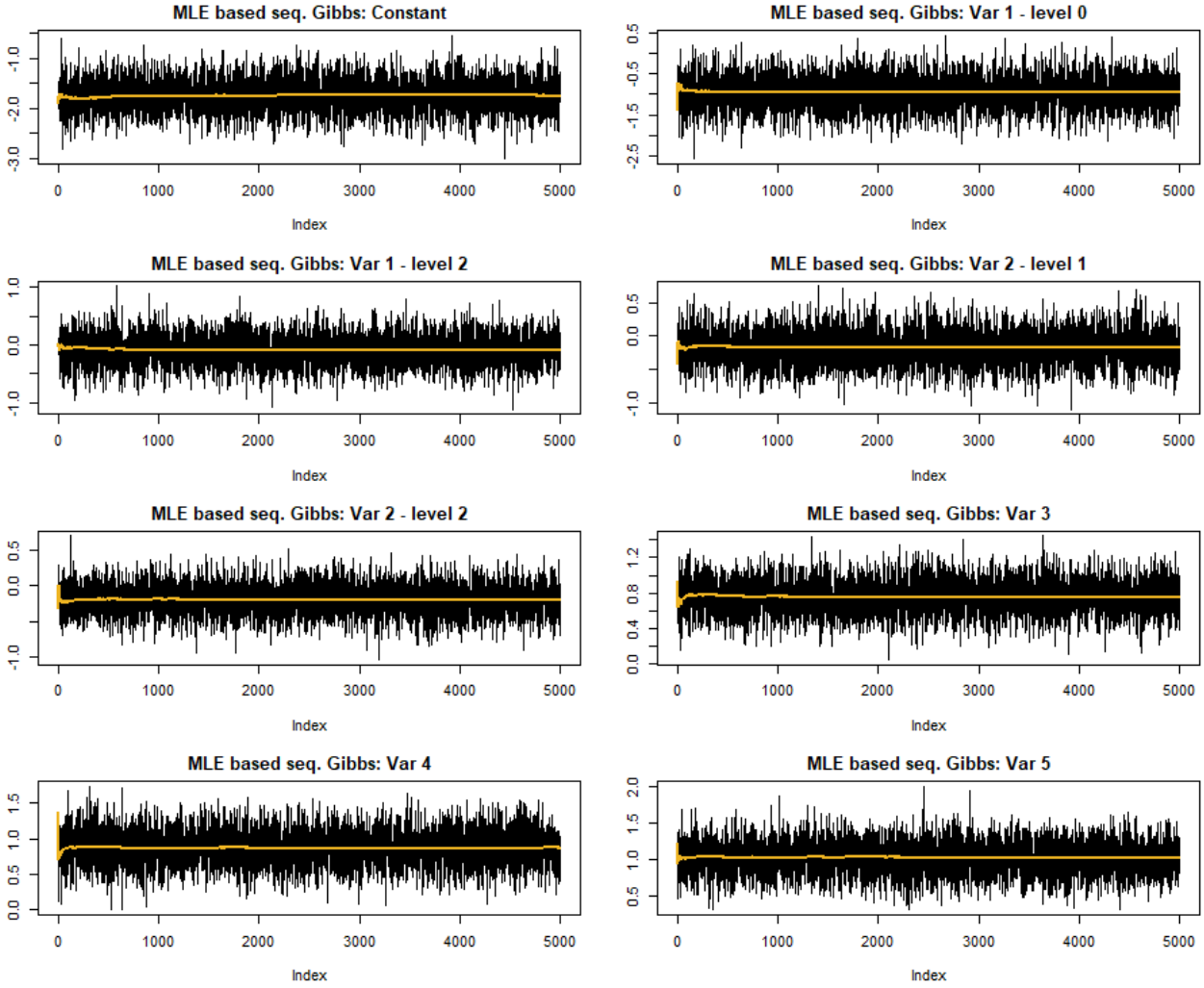
C.1 Markov chain convergence

In order to develop a Bayesian approach for BAIT, we have applied several Bayesian approaches to a binary decision problem. Among these approaches, there are two MCMC methods, which are the Gibbs sampler for the probit model and the Metropolis-Hastings (MH) sampler for the logit model. Important for these MCMC samplers is that the Markov chains in these approaches have converged. This can be verified by plotting the Markov chains and checking whether the sampled β's follow a stationary distribution.

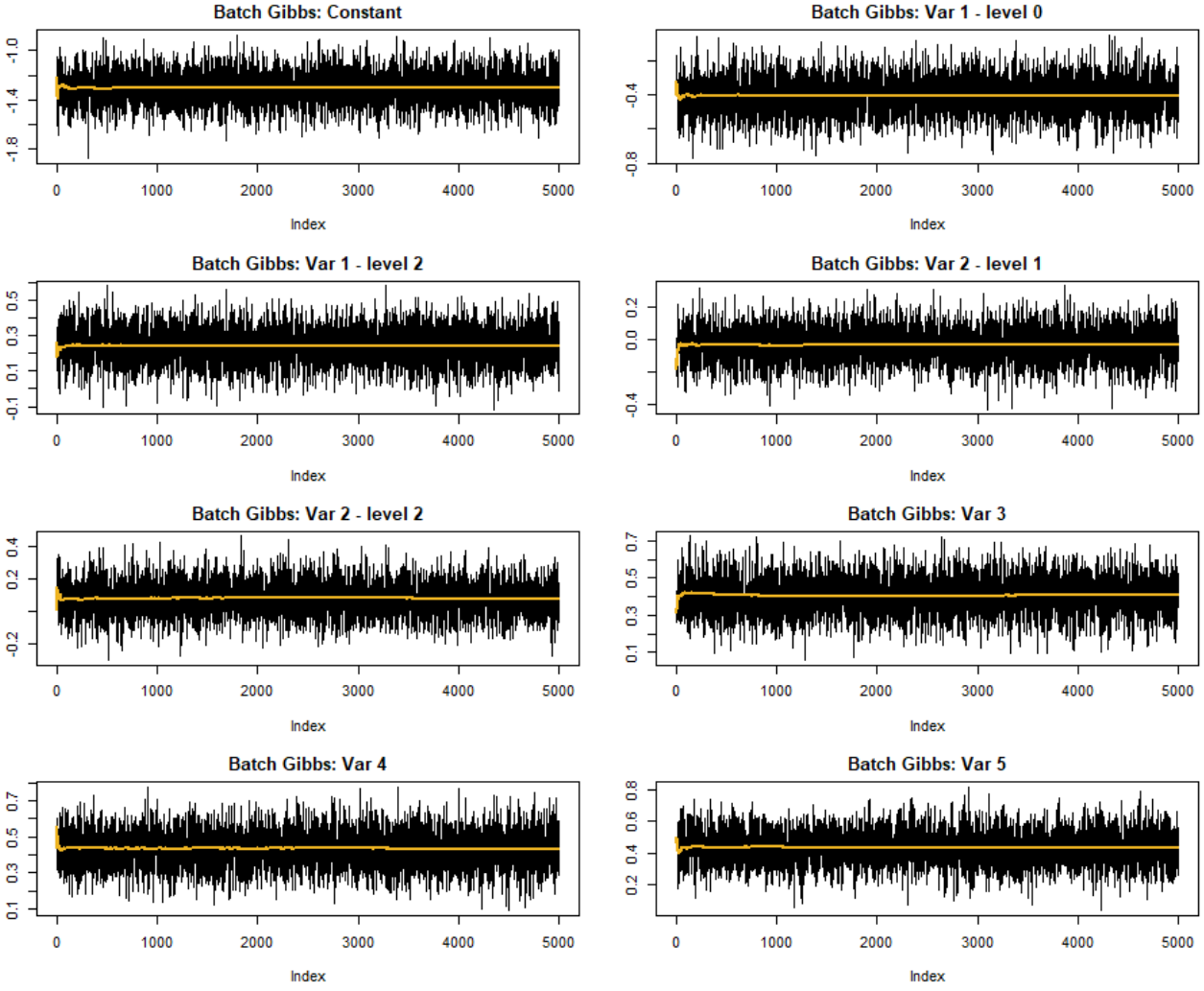
Figure C.1 displays the stored samples of all β_j's for the Markov chains of MCMC models for the special welfare application data set. For the batch Bayesian approaches, we apply one single Bayesian model, which means this is the only Markov chain used to determine the posterior parameter. For the sequential Bayesian approach, we compute a Markov chain at every newly absorbed data point. For these sequential models, we have provided the plot of the Markov chain when the last data point was absorbed. We see for every data point the chain has converged.

Figure C.1: Verification of convergence of the Markov chains for all parameters of every probit Gibbs and logit MH model

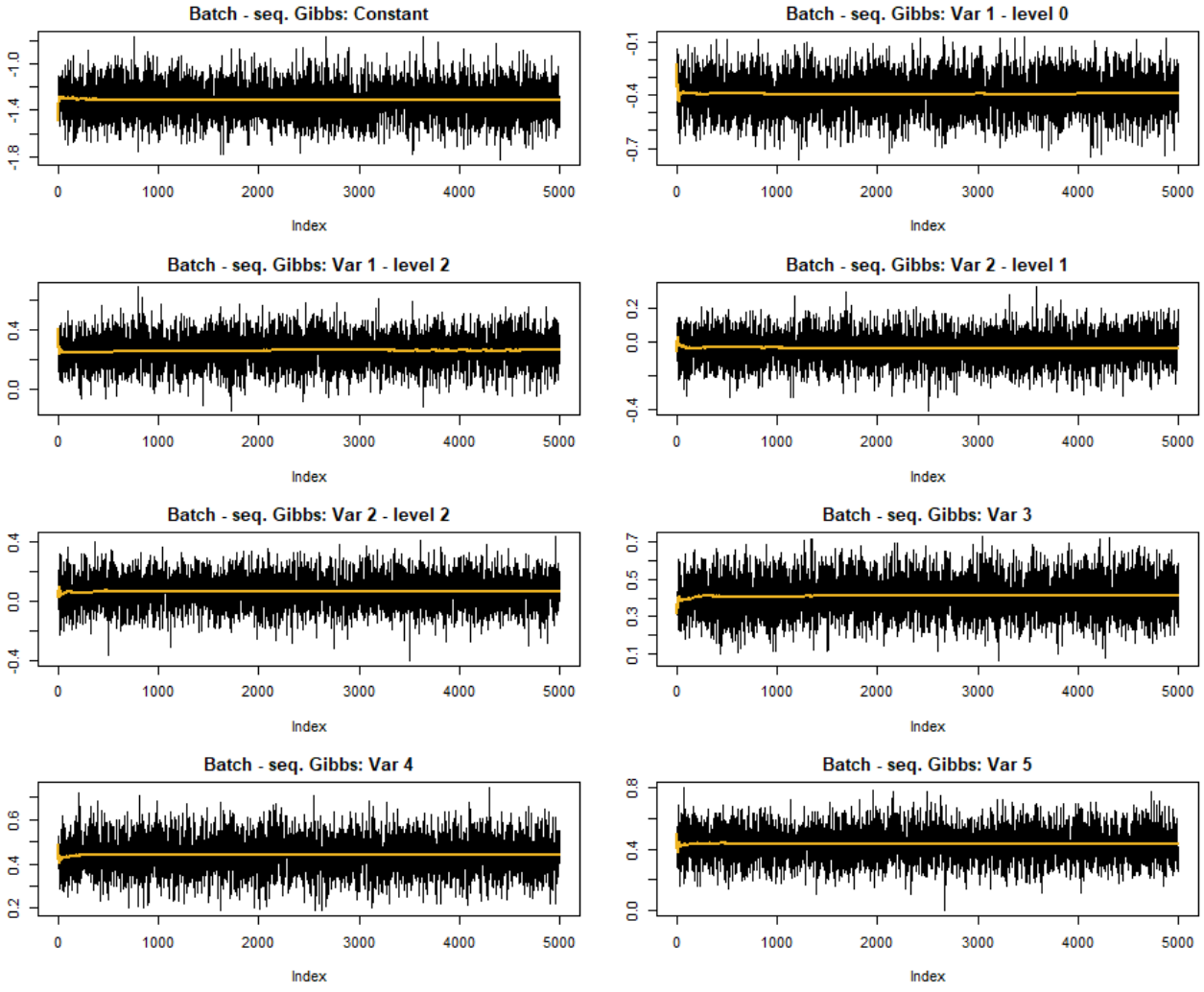
(a) Convergence of Markov chains under MLE-based sequential Gibbs



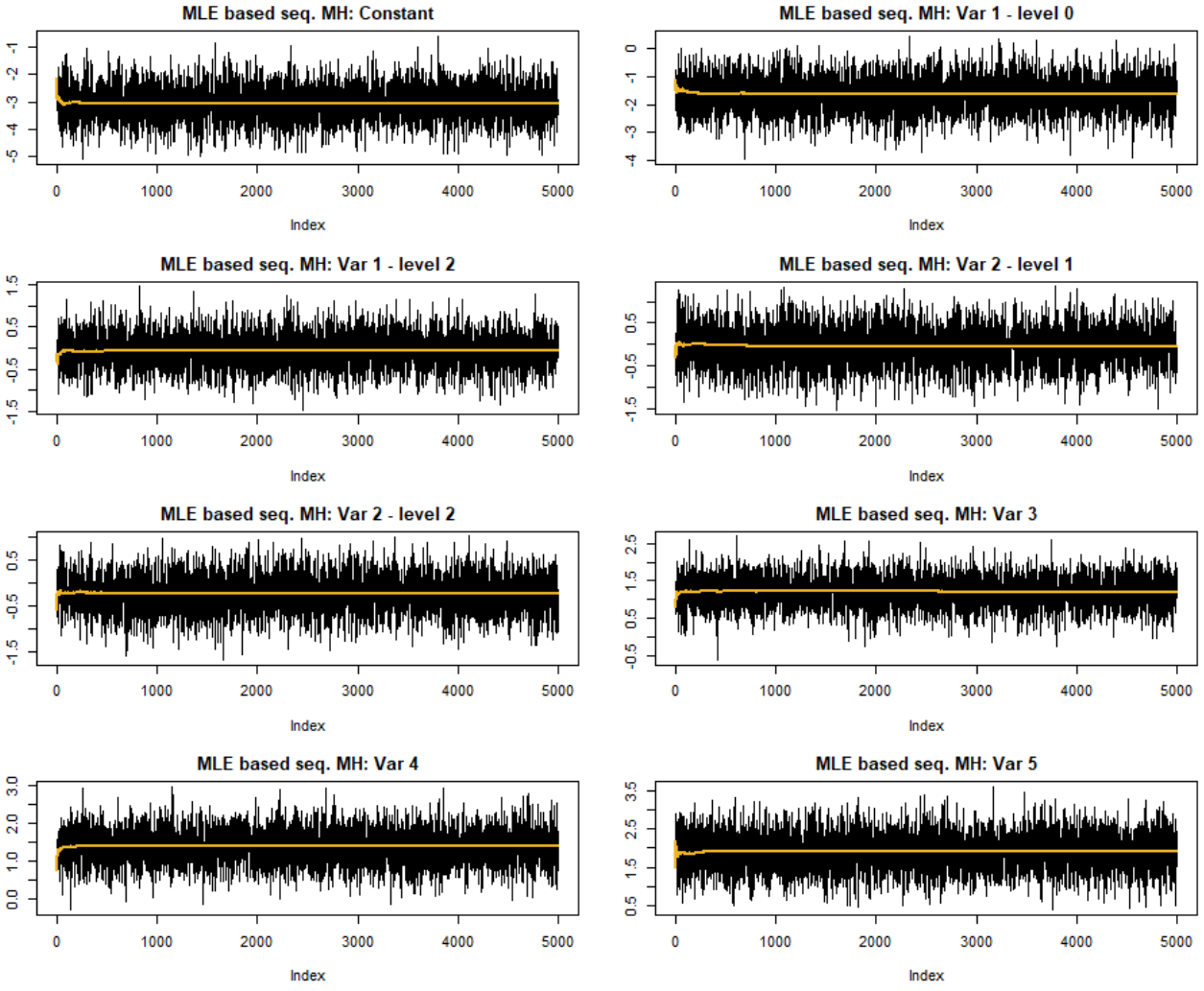
(b) Convergence of Markov chains for Batch Gibbs



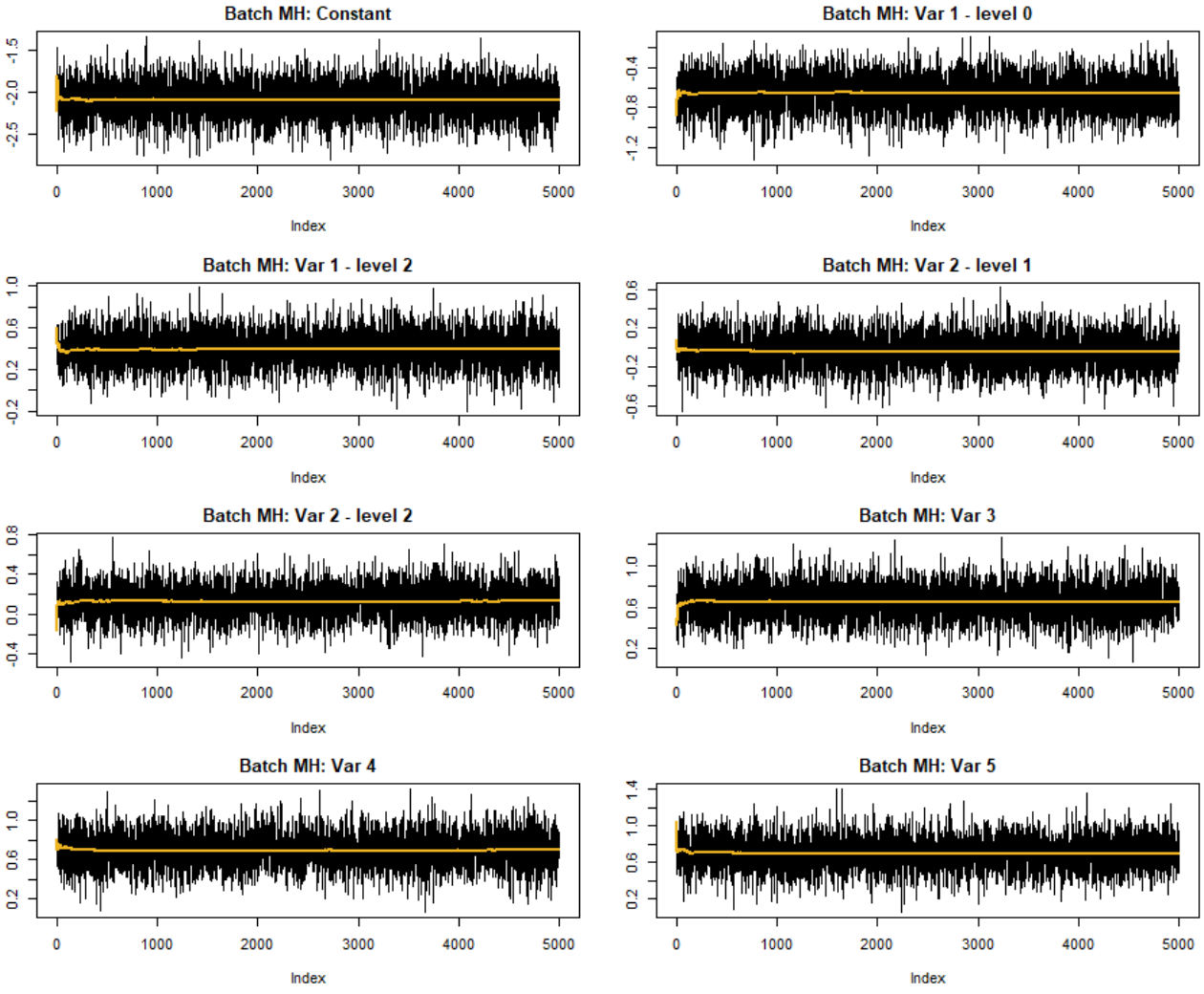
(c) Convergence of Markov chains for Batch - sequential Gibbs



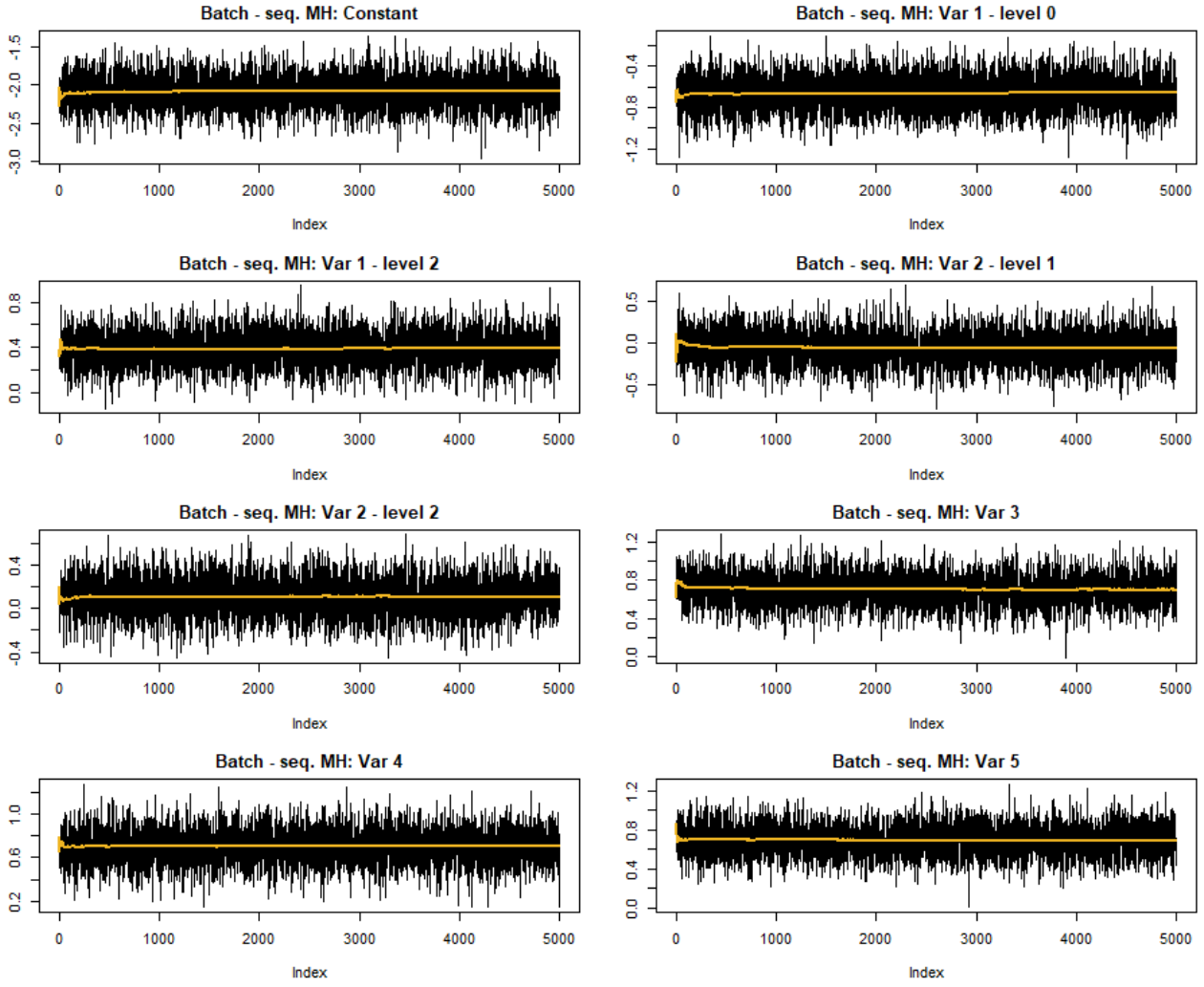
(d) Convergence of Markov chains for MLE-based sequential MH



(e) Convergence of Markov chains for Batch MH



(f) Convergence of Markov chains for Batch - sequential MH



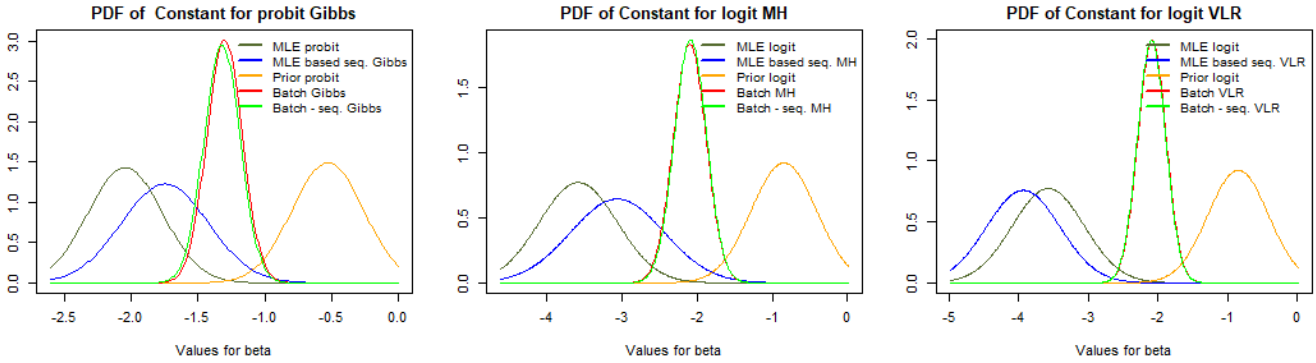
C.2 Comparison of posterior distributions

In Section 4.5, we have discussed the various posterior distributions under different model settings. For completeness, Figure C.2 provides plots of the probability density functions for all parameters in every model. Per plot, we compare the parameters in different model settings for the same Bayesian model.

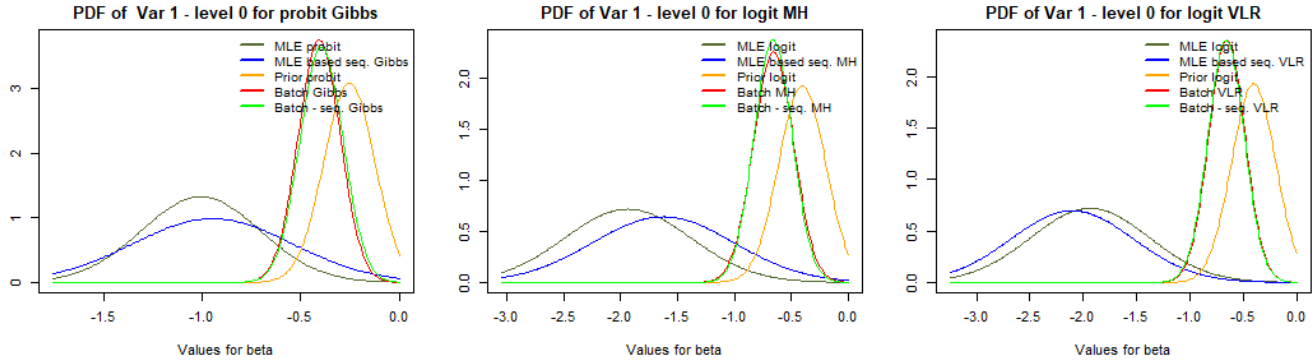
All these plots confirm our general conclusions from Section 4.5. In general, the posterior distributions of the various model settings in the probit Gibbs sampler, logit MH sampler and VLR model show the same trends. The prior is chosen more informative than the parameter estimates in MLE. The MLE-based sequential Bayesian approach closely mimics the MLE results. The batch - sequential Bayesian approach is very close to the batch Bayesian approach, where the only difference is the approximation error in the batch - sequential approach due to sequential approximation of the posterior by a Gaussian distribution.

Figure C.2: Comparison of the posterior distributions of the Bayesian approaches, relative to MLE parameter estimates and the prior densities

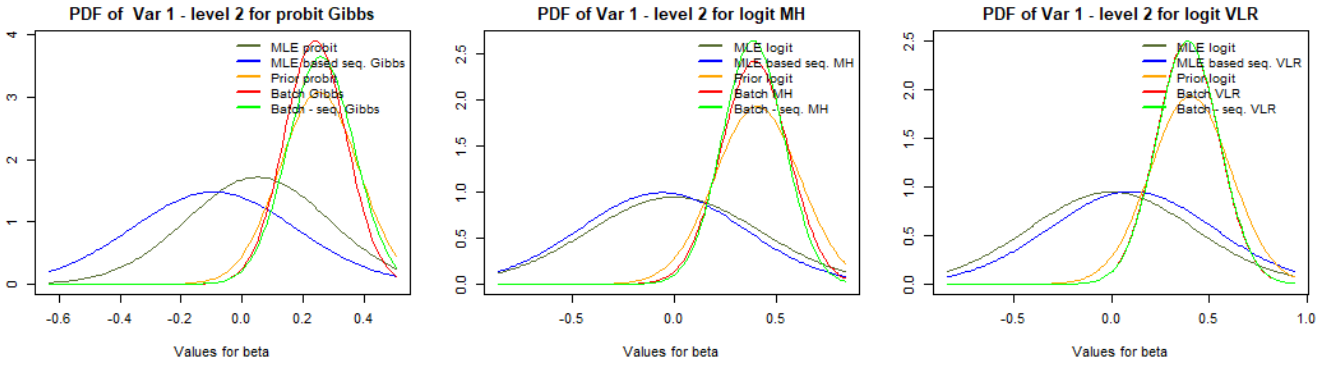
(a) Posterior distributions for the constant



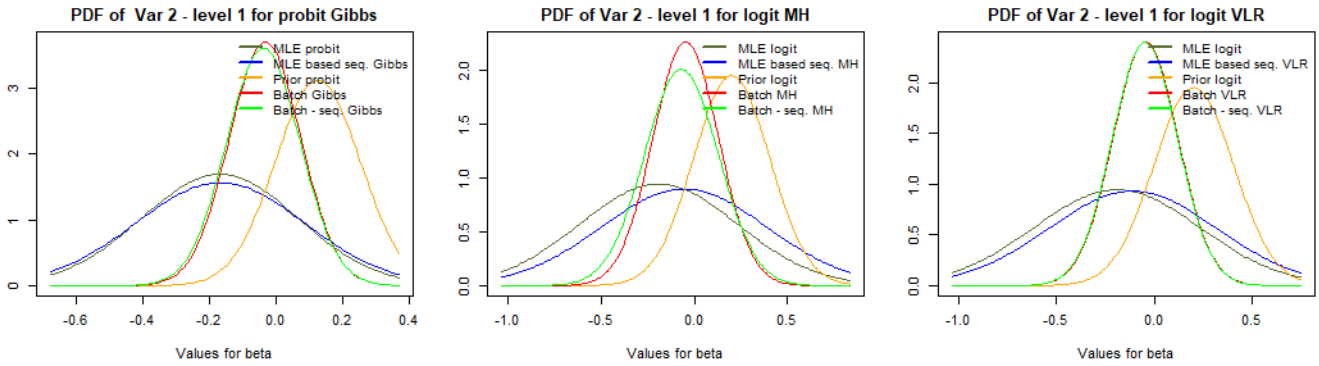
(b) Posterior distributions of the effect of Level 0 of Var 1



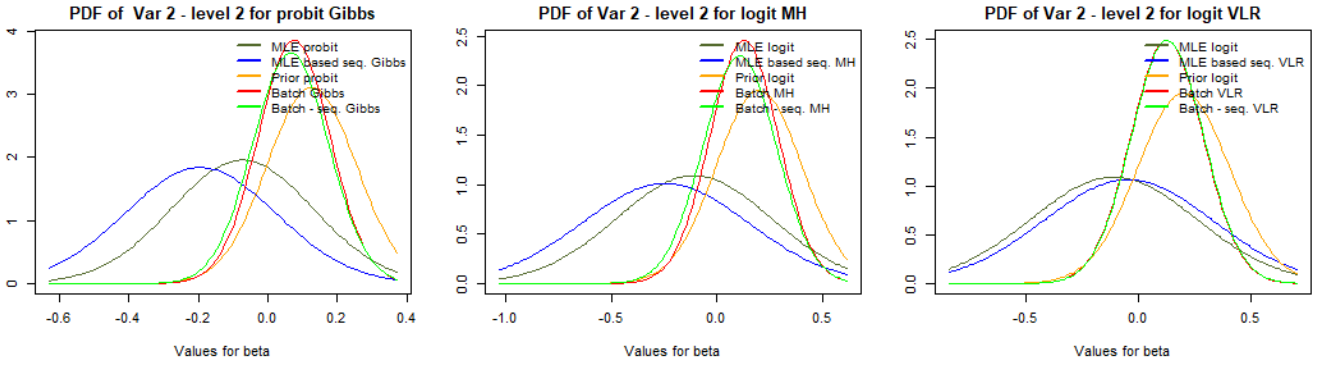
(c) Posterior distributions of the effect of Level 2 of Var 1



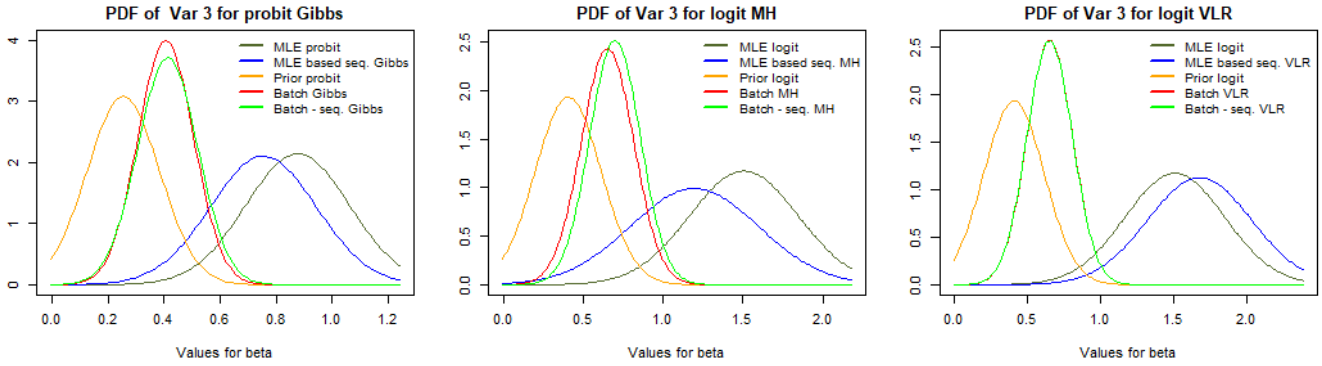
(d) Posterior distributions of the effect of Level 1 of Var 2



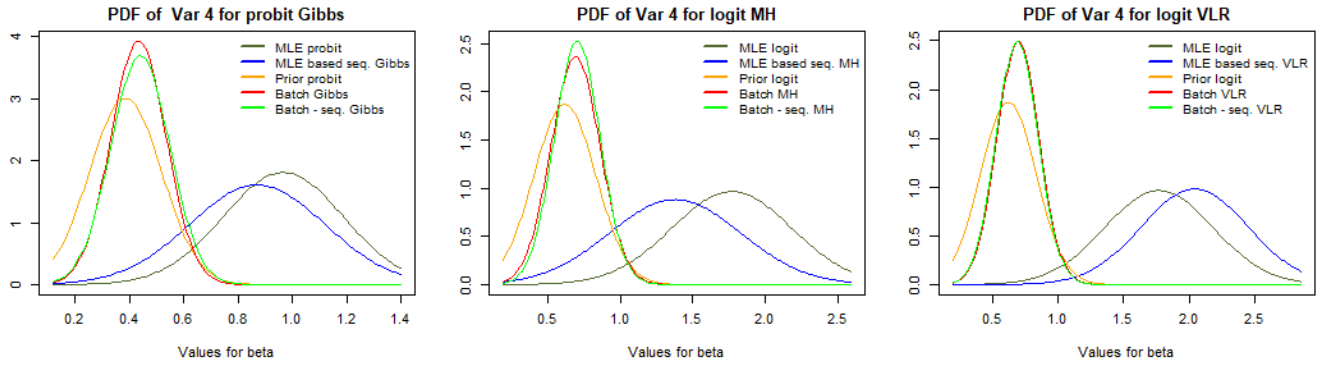
(e) Posterior distributions of the effect of Level 2 of Var 2



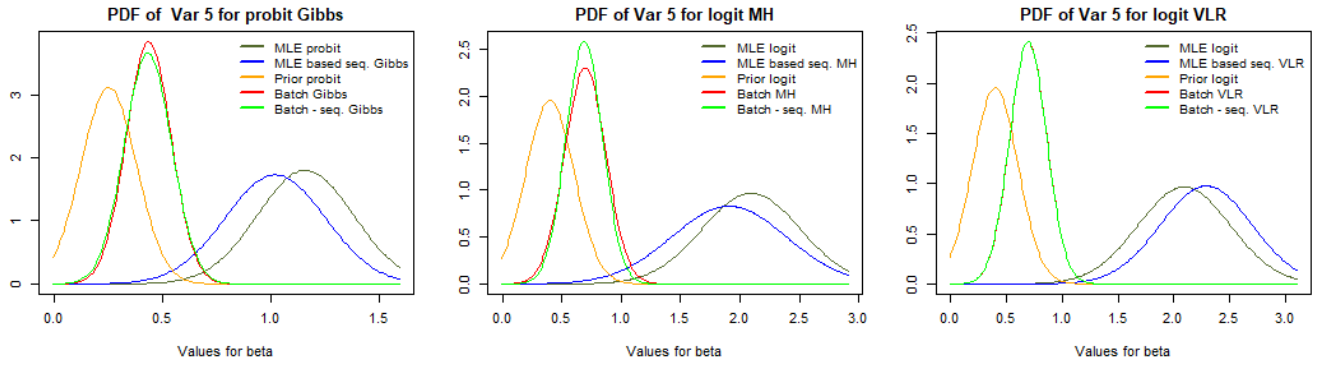
(f) Posterior distributions of the effect of Var 3



(g) Posterior distributions of the effect of Var 4



(h) Posterior distributions of the effect of Var 5

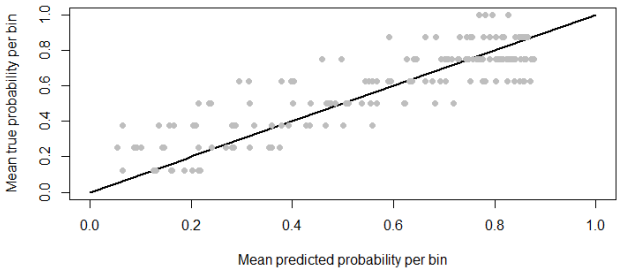


C.3 Calibration plots

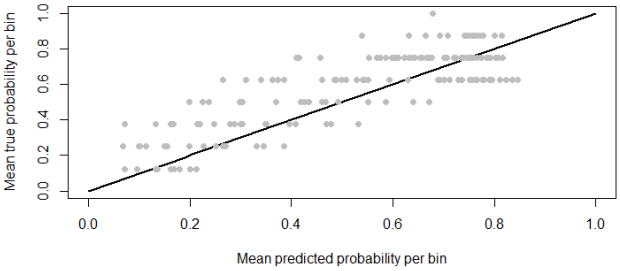
In this section we provide a complete overview of the calibration plots for all models. These plots confirm our conclusions drawn in Section 4.5. The MLE and MLE-based models provide the best calibrated model, as the mean absolute deviation of the grey dots with respect to the ideal black line is smallest. The batch Bayesian and batch - sequential Bayesian approaches provide a worse calibration.

Figure C.3: The calibration plots for all models

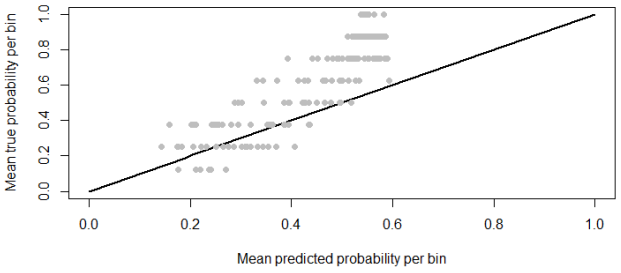
(a) Calibration plot of probit MLE



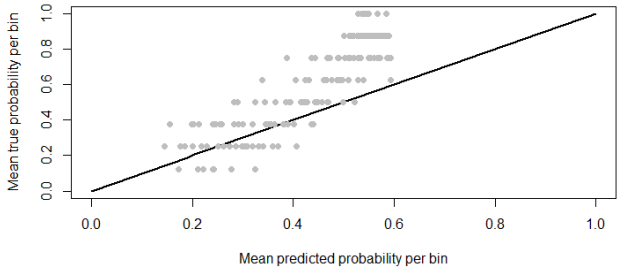
(b) Calibration plot of MLE-based sequential Gibbs



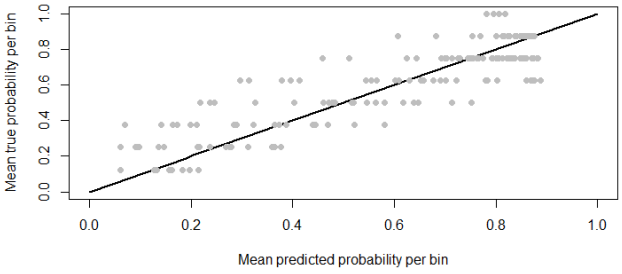
(c) Calibration plot of Batch Gibbs



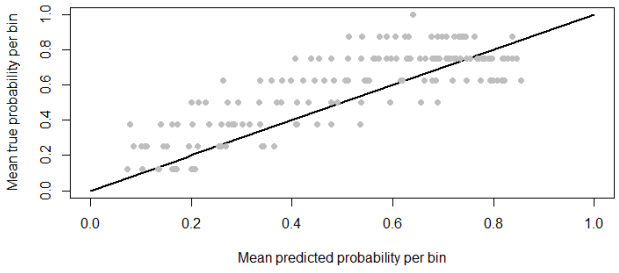
(d) Calibration plot of Batch - sequential Gibbs



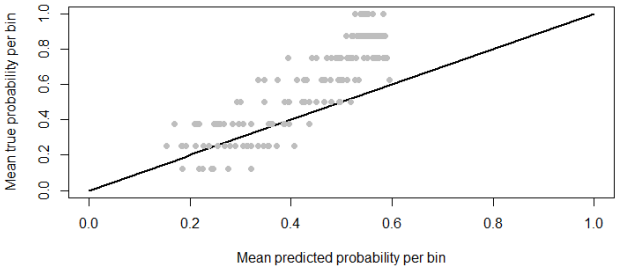
(e) Calibration plot of logit MLE



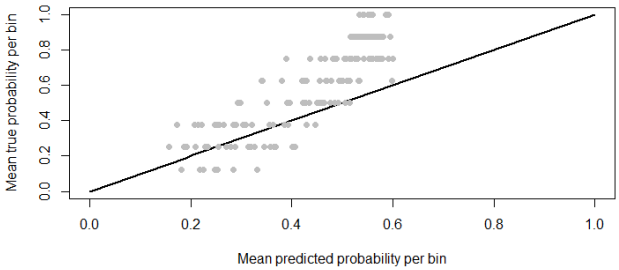
(f) Calibration plot of MLE-based sequential MH



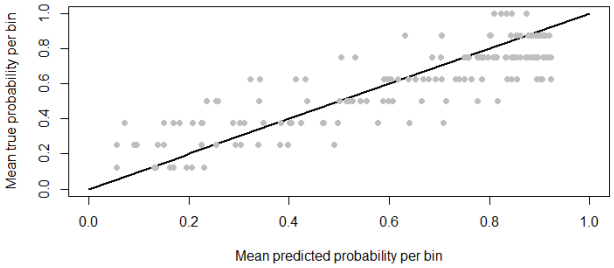
(g) Calibration plot of Batch MH



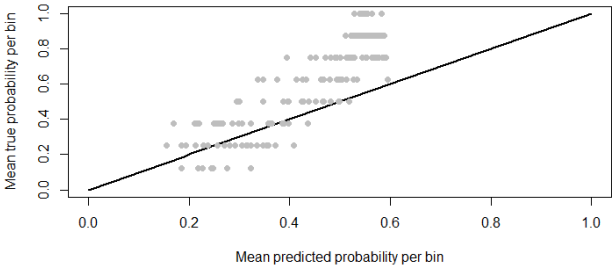
(h) Calibration plot of Batch - sequential MH



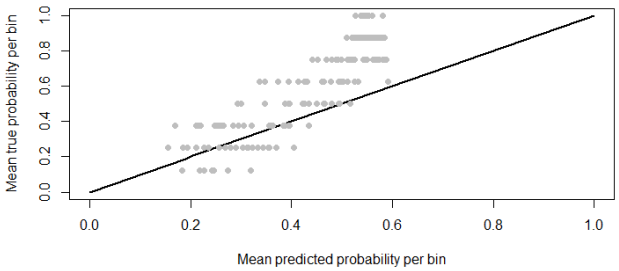
(i) Calibration plot of MLE-based sequential VLR



(j) Calibration plot of Batch VLR



(k) Calibration plot of Batch - sequential VLR



Appendix D

Simulated expert beliefs

D.1 Regular experts

We have simulated 25 regular experts according to $\boldsymbol{\beta}_i^{expert} \sim N(\boldsymbol{\beta}^{MLE}, \kappa \cdot \Sigma^{MLE})$. Here, the scaling factor κ has become 7.2 according to desired average agreement rate of 0.749. The results for the simulated experts can be found in Figure D.1. In the baseline scenario, we assume 10 regular experts: simulated experts 101 to 110. In the scenario for small group sizes, we have taken experts 101 to 103 and in the large group case, we have considered experts 101 to 125.

Table D.1: Simulated beliefs of regular experts

Expert ID	Constant	Var 1 - level 0	Var 1 - level 2	Var 2 - level 1	Var 2 - level 2	Var 3	Var 4	Var 5
101	-4.945	-2.608	-0.354	1.118	-0.379	2.954	3.073	2.858
102	-3.93	-2.032	0.581	0.067	-1.962	1.077	0.663	3.954
103	-7.047	-1.82	1.401	-0.043	1.66	3.225	5.362	1.896
104	-7.919	-2.144	1.594	-1.231	0.78	5.111	5.092	1.605
105	-5.271	-4.859	0.76	-1.748	0.609	2.849	4.869	2.966
106	-3.976	-3.851	1.141	0.465	-0.496	1.656	3.308	2.118
107	-0.794	2.142	0.538	-1.286	-2.476	-1.267	-1.465	0.998
108	-6.928	-4.352	2.003	0.951	0.741	1.995	4.335	3.882
109	-2.502	-3.996	-3.359	1.213	-0.053	-1.206	3.402	2.784
110	-4.459	-3.433	1.427	-0.591	-0.093	3.278	3.034	3.102
111	-8.076	-2.752	1.372	-0.1	1.047	5.162	4.018	3.01
112	-2.935	-4.39	-0.966	-0.608	-1.407	1.88	3.935	2.768
113	-4.165	-2.495	-1.618	-0.169	1.12	2.236	2.18	3.031
114	-8.987	-3.641	3.145	-0.391	1.885	5.243	4.564	3.187
115	-7.113	-5.344	0.838	-0.135	1.858	4.224	4.266	3.004
116	-6.348	-1.849	1.008	0.919	0.259	3.072	2.459	4.26
117	-8.77	-6.83	0.801	-0.082	0.732	3.809	7.053	5.346
118	-9.456	-4.722	2.941	-0.518	0.292	5.763	5.734	4.494
119	-2.493	-3.067	-2.869	1.973	-0.381	1.351	1.487	1.609
120	-4.847	-3.074	0.817	-1.084	-0.013	4.279	2.149	2.757
121	-5.715	-3.858	0.873	0.707	2.077	2.563	4.299	2.184
122	-8.176	-2.596	1.845	1.422	1.873	5.306	4.092	3.022
123	-3.644	-2.559	-0.903	-0.69	0.461	2.851	2.883	0.655
124	-6.502	-4.577	1.905	0.042	3.247	2.688	4.279	2.631
125	-3.012	-5.917	-2.509	-1.587	-0.807	1.679	5.042	3.797

D.2 Opposing experts

We have simulated 10 opposing experts according to $\boldsymbol{\beta}_i^{expert} \sim N(\boldsymbol{\beta}^{MLE}, -\kappa \cdot \Sigma^{MLE})$. Similar to the regular heterogeneity case, κ is set to 6.5. The results can be found in Figure D.2. In the scenario with 1 opposing expert, we take consider expert 201 as the opposing expert. In the scenario with two groups of 5 opposing experts, we consider experts 201 to 205.

Table D.2: Simulated beliefs of opposing experts

Expert ID	Constant	Var 1 - level 0	Var 1 - level 2	Var 2 - level 1	Var 2 - level 2	Var 3	Var 4	Var 5
201	5.426	3.711	0.118	-0.142	0.366	-3.272	-4.734	-2.792
202	6.535	1.664	-2.44	0.106	-0.788	-3.379	-2.557	-2.783
203	5.619	1.807	-1.754	0.161	-0.323	-2.684	-1.95	-2.156
204	8.446	2.099	-1.939	-0.864	-1.68	-3.903	-2.835	-4.005
205	3.449	5.414	1.077	0.256	-0.836	-1.83	-3.987	-2.218

D.3 Homogeneous experts

We have simulated 10 homogeneous experts according to $\boldsymbol{\beta}_i^{expert} \sim N(\boldsymbol{\beta}^{MLE}, \kappa_{hom} \cdot \Sigma^{MLE})$. Scaling factor κ_{hom} has become 1.9 according to desired average agreement rate of 0.9. The results for homogeneous experts 301 to 310 can be found in Figure D.3.

Table D.3: Simulated beliefs of homogeneous experts

Expert ID	Constant	Var 1 - level 0	Var 1 - level 2	Var 2 - level 1	Var 2 - level 2	Var 3	Var 4	Var 5
301	-6.242	-3.641	0.636	0.028	0.994	2.577	4.445	3.317
302	-7.055	-2.9	1.128	-0.352	0.671	2.952	4.695	3.134
303	-5.961	-3.282	1.286	-0.253	1.026	3.109	4.168	2.467
304	-6.151	-3.653	0.839	1.263	1.616	3.153	3.053	3.112
305	-7.354	-4.145	1.522	0.428	0.707	3.37	4.798	4.135
306	-3.443	-2.843	-0.706	0.376	1.267	1.868	2.297	1.794
307	-5.096	-3.662	0.805	0.54	0.907	2.21	3.322	2.836
308	-5.844	-2.38	1.608	1.348	1.558	3.051	2.125	1.925
309	-7.408	-2.293	1.416	-0.306	1.086	2.566	3.631	4.05
310	-7.04	-3.834	1.005	0.356	-0.092	3.841	3.724	3.223

D.4 Heterogeneous experts

We have simulated 10 homogeneous experts according to $\boldsymbol{\beta}_i^{expert} \sim N(\boldsymbol{\beta}^{MLE}, \kappa_{het} \cdot \Sigma^{MLE})$. Scaling factor κ_{het} has become 51 according to desired average agreement rate of 0.5. The results for heterogeneous experts 401 to 410 can be found in Figure D.4.

Table D.4: Simulated beliefs of heterogeneous experts

Expert ID	Constant	Var 1 - level 0	Var 1 - level 2	Var 2 - level 1	Var 2 - level 2	Var 3	Var 4	Var 5
401	-9.398	-1.107	3.225	2.583	2.291	5.665	8.402	1.547
402	4.51	-1.639	0.315	-1.329	-1.553	-4.323	-4.042	1.019
403	-12.522	-8.614	2.342	2.264	3.375	4.934	8.943	6.698
404	-5.422	0.991	3.63	-0.627	1.042	2.687	2.146	2.276
405	-10.211	-2.63	4.893	3.619	1.956	4.315	6.917	3.281
406	-2.682	-4.484	0.101	-1.345	3.085	2.736	-1.371	4.475
407	5.529	10.021	-0.291	-6.448	-1.8	-4.408	-5.245	-4.137
408	-5.889	-6.929	1.34	-2.476	1.893	3.955	5.828	0.228
409	-2.199	0.091	-1.764	3.955	3.889	1.313	-0.951	-0.564
410	-1.672	-2.844	-4.262	4.871	-2.047	2.26	5.362	2.913