MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

A Bayesian odyssey: A psychologist's view on the use of (Bayesian) computational models in cognitive science

verfasst von / submitted by

Fabian Marvin Renz, BSc MSc

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien, 2022 / Vienna 2022

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	UA 066 013
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Masterstudium Joint Degree Programme MEi: CogSci Cognitive Science
Betreut von / Supervisor:	Univ.-Lektor Dipl.-Ing. Dr. Paolo Petta

Contents

1. Introduction to Computational Modelling in Cognitive Science

"If you test your programs not merely by what they can accomplish, but how *they accomplish it, they [sic!] you're really doing cognitive science; you're using AI to understand the human mind."* (Herbert Simon, Stewart 1994)

Stewart D. (1994). Interview with Herbert Simon. *Omni Magazine*, June 1994.

1.1 The purposes of models in Cognitive Science

Cognitive scientists seek to *understand* cognition: They want to describe, predict and ultimately be able to explain it. Various scientific disciplines have developed and refined methodological approaches to study cognition. One dominant approach that evolved across disciplines including psychology, artificial intelligence or neuroscience is to build *models* of cognition. The Stanford Encyclopedia of Philosophy identifies 30 different types of models serving different purposes in science (Frigg & Hartmann, 2020): These include representational models, phenomenological models, computational models, explanatory models and exploratory models among many others. "*While at first glance this abundance can be overwhelming, it can be brought under control by recognizing that these notions pertain to different problems"*, whereas the problem refers to the phenomenon of interest. (Frigg & Hartmann, 2020). Furthermore, these categories are not mutually exclusive, and a given model can fall into several categories at once.

Many scientific models are representational models: They represent a selected aspect of the world, referred to as the model's target system (Frigg & Hartmann, 2020). A model represents its target system in a simpler, more abstract form. Generally, representational models are applied to their target system with the aim to increase the understanding of the modelled system (Frigg & Hartmann, 2020). They provide the researcher with the means to approach a phenomenon too complex or too difficult to deal with directly (Fum et al., 2007): A model corresponds to a simplified and more abstract version of the system while keeping the essential features and omitting unnecessary details. The insights generated by studying and experimenting with the model can then be applied back to the original system to increase its' understanding.

Farrell and Lewandowsky (2018, p.3) take the stand that cognitive scientists *must* rely on *quantitative* models to expand their knowledge about the human mind, since verbal theorizing is insufficient on its' own. When relying on pure verbal reasoning and mental simulation scientists are prone to make unjustified assumptions or omit relevant aspects.

Furthermore, they point out that collected data can never speak for itself but require a model to be understood and explained (Farrell & Lewandowsky, 2018, p.6). To understand a phenomenon there are always multiple models that can be applied to the data to explain it. The task is to find the model which explains the data "best". This process, called *model selection*, rests on both quantitative analysis and intellectual, scholarly judgement (Farrell & Lewandowsky, 2018, p.6). What makes a good model and according to which criteria models are evaluated and compared will be addressed in chapter 1.3 of this thesis.

One decisive reason for the usage of *quantitative computational* modelling is the necessity to explicitly state all contributing variables. This allows for matching and evaluating a model against the collected data and to eventually choose the most appropriate model out of many competing models. For this reason, quantitative models are often referred to as a *cognitive aid* for scientists that protect the scientist from their fallibility by pointing out unjustified assumptions and highlighting shortcomings of pure verbal reasoning and mental simulation (Farrell & Lewandowsky, 2018, p.22; Lewandowsky & Oberauer, 2018, p.5).

Coming from a classical training in psychology, a discipline with the overarching goal to understand the human mind and behaviour (VandenBos, 2007), my first personal motivation for this thesis is to explore the additional angle computational models offer. Psychology in the recent past faced multiple problems. In my personal opinion, two of the most pressing ones are on the one hand, the replication crisis (Fidler & Wilcox, 2021; Wiggins & Chrisopherson, 2019), and on the other hand, the question of how to construct and test theories of cognitive mechanisms that underlie the observed behaviour (Guest & Martin, 2021). Both of these problems can be addressed to some extent by the application of computational models (Farrell & Lewandowsky, 2018; Oberauer & Lewandowsky, 2019). While computational models are regularly advocated in building and testing theories of cognition, the philosophical status of modelling per se often receives less attention compared to details of specific models (Stafford, 2012). Thus, the second motivation for this thesis is to explore the role of computational models and how they are scientifically useful for cognitive science.

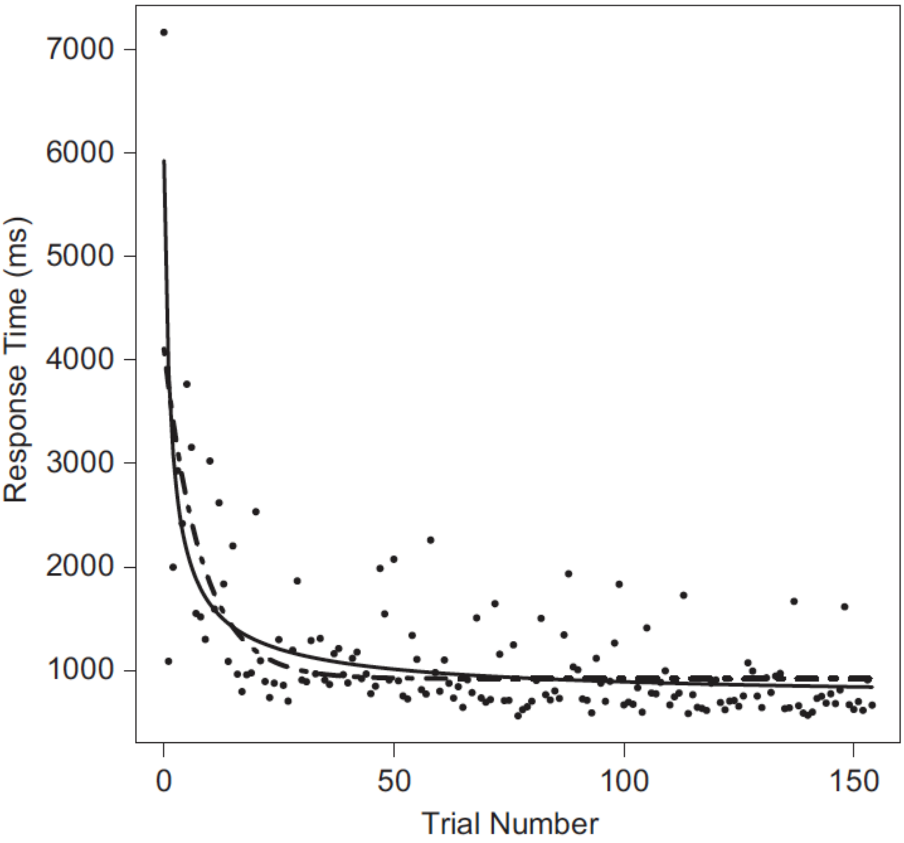
Computational models of cognition are specified at different levels of explanation, at varying levels of granularity (Lewandowsky & Oberauer, 2018). Different taxonomies and potential alternatives will be described first as well as important distinctions, followed by a more thorough explanation of Bayesian methods for cognitive modelling and their contributions to cognitive science. Following the introduction of Bayesian methods the focus in the second half of the thesis is centered around a case study lent from delay discounting research (Mazur, 1987), demonstrating the core principles of Bayesian cognitive modelling.

1.2 Statistical Data description and Cognitive Process Models

An important distinction between different model types is between *descriptive models* and *cognitive process models*. Descriptive models of cognition make no claims about underlying cognitive processes but capture empirical regularities. They describe a cognitive phenomenon as for example the acquisition of a new skill (Farrell & Lewandowsky, 2018, see Figure 1).

Figure 1

Skill Acquisition Modelled as Power Law or Exponential Improvement



Note. Performance models describing skill acquisition. The data from a skill acquisition task shows the exponential decay in response time with an increase in trial number. A power law function (solid line) and an exponential function (dashed line) (Heathcote et al., 2000) are fitted to the data from Palmeri (1997). Taken from Farrell & Lewandowsky (2018, p.11).

Figure 1 shows two models aimed at capturing the decay in response time with increasing number of trials. The “Power model”, which has been prevailing for a long time is given by the function 1.1.

$$RT = N^{-\beta}. \qquad 1.1$$

Here, RT represents the reaction time, while N gives the number of trials and β indicates the learning rate. However, Heathcote et al. (2000) argued that the data are better described by an exponential function, which in its simplest form is given by the function 1.2.

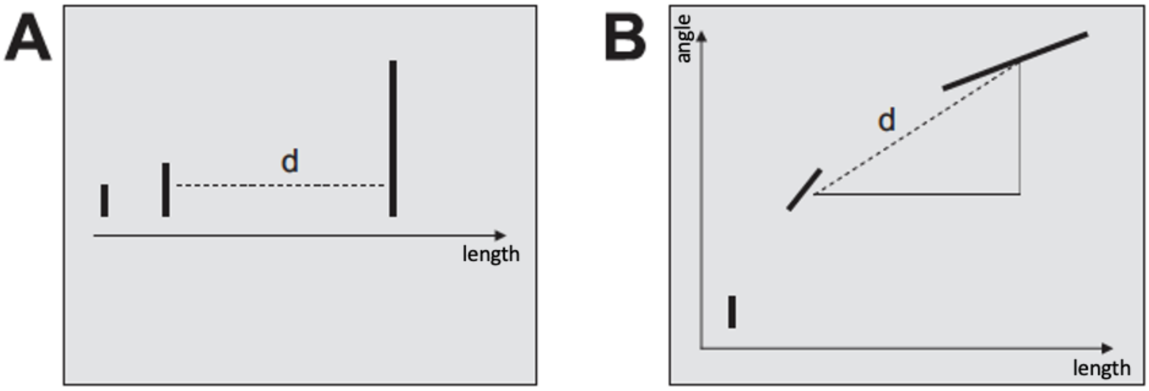
$$RT = \mathrm{e}^{-\alpha N}. \qquad 1.2$$

Here, N is as before in 1.1 and α represents the learning rate. The learning rate can be thought of as a scaling parameter, affecting the influence of the number of trials on the reaction time. Both models in Figure 1 are fitted to the data of a study originally conducted by Palmeri (1997). The models only describe the relation between practice and decrease in reaction time, but do not make any claims about the underlying cognitive mechanisms. However, it is important to note that although the models do not make any claims about the underlying cognitive mechanisms, the choice of descriptive model carries implications. In this case, it carries implications about “*the nature of learning”* (Farrell & Lewandowsky, 2018, p.11). In the present example, the two models have different implications regarding the learning rate. The exponential model implies that the learning rate relative to what remains to be learned is constant throughout practice. The power law implies that the relative learning rate slows down as practice increases. For a more detailed explanation of the two models and their comparison see Heathcote et al. (2000).

Cognitive process models move beyond a pure description of statistical regularities and represent theories about the underlying mechanisms producing the observable behaviour. Farrell and Lewandowsky (2018) introduce the example of the Generalized Context Model (GCM; Nosofsky, 1986). The GCM is used to describe the categorization of stimuli according to their similarities as indicated by their representative distances. The representative distance is a measure quantifying the similarity along one or multiple dimensions. Psychological distances can be thought of similarly to more familiar measures such as for example a spatial distance being measured in centimeters. Generally, low representative differences indicate a high similarity, and a high distance indicates low similarity. Examples visualizing the representative distance are given in Figure 2.

Figure 2

Representative difference example



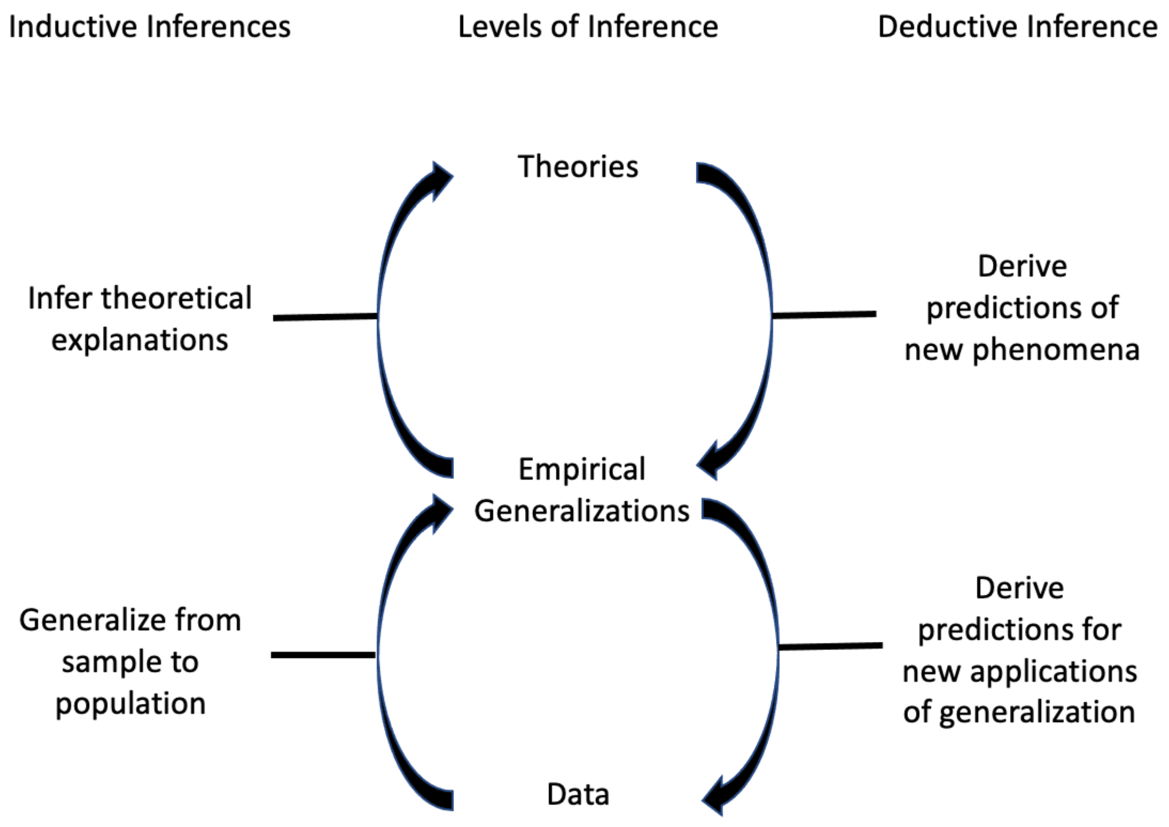
Note. An example illustrating the representative difference. In panel A the three lines only differ along one dimension the line length. In panel B the lines differ along two dimensions, namely line length and angle. The dashed lines in both panels show the representative distance (d). Adapted from Farrell & Lewandowsky (2018, p.14 Figure 1.6).

In the GCM the representative distance is the basis for the activation of a specific *category*, which subsequently determines the *category assignment*. As a result, different stimuli with a low representative distance are grouped together into the same *category*. The GCM for categorization can also be applied to richer stimuli than the simple line example from Figure 2. For a detailed explanation of the GCM and representative distances see Nosofsky (1986, 2011). Here the emphasis is not on the cognitive phenomenon and the attempt to capture it in a model, but rather on the difference between process models and descriptive models. The GCM model as an example process model does not only aim at describing the relation between the stimuli characteristics and the behaviour (e.g., assigned category) but presumes that the computations specified in the model represent an analogue process used by people to solve the task. According to the GCM in order to assign an object to a given category a representative distance is computed resulting in a category assignment. Thereby, the model represents a theory of how people solve the task of categorization and the underlying mechanisms.

The distinction between descriptive models and process models can be linked to the difference between statistical and cognitive models. Lewandowsky & Oberauer (2018) point out two levels of inference scientific reasoning builds upon (Figure 3). The first level of inference establishes links between data and empirical generalizations. Here, inductive inferences are made from data collected in individual studies to empirical generalizations. These generalizations are statements that "*pertain to a regular relationship between observable variables"* (Lewandowsky & Oberauer, 2018, p. 1). At the same level, the established or hypothesized generalizations are used for deductive inferences to predict the outcomes of future studies.

Figure 3

Levels of Inference in Scientific Reasoning


Inductive Inferences
Levels of Inference
Deductive Inference
Theories
Infer theoretical explanations
Derive predictions of new phenomena
Empirical Generalizations
Generalize from sample to population
Derive predictions for new applications of generalization
Data

Note. Levels of inference in scientific reasoning: The left column displays *inductive inferences*. Based on Data Empirical Generalizations are drawn. From these generalizations Theories can be inferred. The right column illustrates *deductive* inferences. Rooted in Theory, Empirical Generalizations are derived from which in turn predictions about new applications and data are obtained. Adapted from Lewandowsky & Oberauer (2018, p. 2 Figure 1.1).

At the second level, empirical generalizations are linked to theories. Theories go beyond empirical generalizations in making assumptions about unobservable variables, mechanisms and their connections to observable variables (Lewandowsky & Oberauer, 2018): Inductive reasoning is used to infer theoretical constructs from empirical generalizations. These theoretical constructs are again used to deductively derive predictions of empirical regularities. Both levels of inference are essential in cognitive science research. However, one aspect in which they differ is their degree of formalization. The first level inference from data to generalizations has increasingly formalized to reduce bias and ambiguity in the inferential process (Cramer et al., 2016; Wagenmakers et al., 2016). At the second level, the investigation of underlying mechanisms by developing theories is less homogenous. Several research areas in cognitive science engage in quantitative computational modelling, e.g., psychophysics or decision making research; in others verbal reasoning has retained a prominent role (Lewandowsky & Oberauer, 2018, p.22). These latter theories based on verbal theorizing and mental simulations are insufficient because they do not necessarily *force* the scientist to make their underlying assumptions explicit (Farrell & Lewandowsky, 2018). This includes explicating *all* relevant variables, with their impact on the outcome. Furthermore, purely verbally formulated models cannot be turned into testable predictions (Lewandowsky & Oberauer, 2018, p.6). One solution to the mentioned limitations is to complement verbal theorizing with quantitative, computational models. In addition to enabling model-based evaluation of the theories, they help communicating (e.g. by sharing model code) and hence are "more" testable and falsifiable (Lewandowsky & Oberauer, 2018, p.21).

The method of falsification is very influential in cognitive science and originally dates back to Popper and the Logical Positivists' (Cooper, 2007; Popper, 1959). It describes the position that *"disproving, rather than proving, of hypotheses is the basic procedure of scientific investigation and the chief means by which scientific knowledge is advanced."* (VandenBos, 2007). Thus, it is essential for a model to be testable and therefore falsifiable. The importance of falsifiability equally applies to computational models of cognition and is essential for computational models next to the possibility of evaluating and comparing models which will be the focus of the following chapter (Palminteri et al., 2017).

1.3 Evaluating and comparing models

There is always more than just one model to explain a phenomenon (Anderson, 2013). When a model fits the data, this implies that the model is *sufficient for the particular dataset at hand*. However, it does not follow that the model is also *necessary*, referring to the possibility of another model explaining the observed behaviour equally well, if not better. The fact that there always is an unknown number of alternative models may appear to be a major limitation of modelling. Lewandowsky & Oberauer (2018) argue, however, that this does not prevent cognitive scientists from working and evaluating the known models, selecting the best among them.

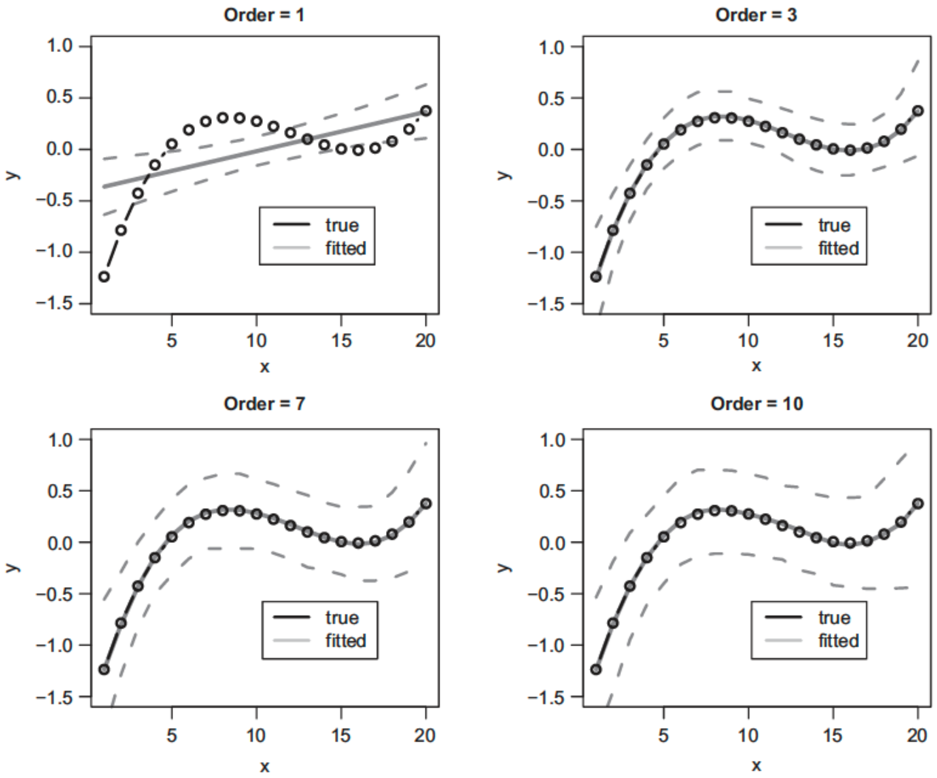
One option to evaluate models is based on their fit of the data. The fit of a model to the data is often quantified using an error function which will be introduced thoroughly in the following chapter 2.4. In short, the error function is a quantification of how well the collected data and the predictions of a model align. However, if a model is only evaluated according to its' fit to the data an important aspect is omitted, namely the so-called *bias–variance trade-off*. This trade-off relates the model's *goodness-of-fit* to the model complicatedness, i.e., the number of parameters in the model. Figure 4 pictures an illustration of the bias–variance trade-off taken from Farrell & Lewandowsky (2018, p.246).

The best model is parsimonious, establishing a balance between *goodness-of-fit* and complicatedness. A too simple model, i. e., with too few parameters, is not able to fit the data, which also referred to as a high bias (Farrell & Lewandowsky, 2018, p.245). A model with too many parameters is too complicated and might overfit the data, also referred to as high variance (Farrell & Lewandowsky, 2018, p.245). Thus, the goal in creating and fitting a model is to build a model including as few parameters as possible i.e., as needed to fit the data (Farrell & Lewandowsky, 2018, p.258).

To assess proposed models in terms of their fit to the data while taking their complicatedness into consideration, several quantitative criteria have been developed. Farrell and Lewandowsky (2018) introduce the Akaike information criterion (AIC) and the Bayesian information criterion (BIC) (Akaike, 1998; Bozdogan, 2000; Schwarz et al., 1978). Introducing additional parameters will reduce deviance a goodness-of-fit metric, but at the same time increase the weight of the penalty term. A means to penalize the inclusion of additional parameters. Thereby, both of them enforce the principle of parsimony aiming at finding the best, simplest model (Myung & Pitt, 2018). When evaluating and comparing models, these criteria are metrics, representing the probabilities of each model being the best model in the set of compared models in the light of the collected data.

Figure 4

Bias – Variance trade off



Note. Illustration of the bias-variance trade-off: In each panel the true generating model – a third order polynomial - is plotted (circles). The four different panels display the fits of polynomials of different orders. The dark grey line corresponds to the average prediction for each model, and the dashed grey lines indicate the variability across simulated datasets. As the model order increases, i. e., the models become more complex, the bias decreases, while the variance increases. Taken from Farrell & Lewandowsky (2018, p. 246).

1.3.1 Narrow and broad models of cognitive phenomena

While a good quantitative fit of a model is a strong argument in favour of a model, it is not the only metric by which a model can and should be evaluated. It can occur that a model reproducing a number of findings across many different experimental paradigms in a qualitative fashion is more valuable than a model which produces precise predictions on a narrow task (Lewandowsky & Oberauer, 2018). This can be the case, although the broader model might not be able to provide the same quantitative fit to the data. While the comparison of narrow models against each other uses well established methods, such as the mentioned quantitative criteria

AIC and BIC, the trade-off between high accuracies on narrow tasks and models aimed at explaining a broader set of experimental paradigms in a qualitative fashion is equivocal.

The comparison between models of differing scale and the arising questions become evident in the memory research example by Farrell & Lewandowsky (2018). In recognition memory research, a subfield of memory research, participants must discriminate between target and lure items, such as items that were previously studied and items that are new and previously unseen (Yonelinas & Parks, 2007). The simplicity of these experimental designs has made them one of the cornerstones in memory research. The participants' performance is often described using a receiver-operator curve that plots correct responses, also referred to as "hits", against false alarms. Although recognition memory being only a small subfield in memory research there is a variety of proposed models including signal-detection models, high-threshold models and dual-process models (Bröder & Schütz, 2009; Wixted, 2007; Yonelinas & Parks, 2007).

Next to these narrow models that rival among each other there are more holistic models of memory attempting to capture a variety of memory research findings across paradigms. For example, the REM model or the temporal-clustering and sequencing model (Farrell, 2012; Shiffrin & Nobel, 1997). The temporal-clustering and sequencing model is a model of short-term and episodic memory stating that people parse their continuous experience into episodic clusters, which are stored as episodes by binding information with an episode to a common temporal context (Farrell, 2012): It is constructed as an attempt to account for a variety of findings in memory research in a qualitative way. Thus, there is the inherent difficulty in comparing narrow models to these broader models, as they are constructed with a different goal in mind. Furthermore, the comparison among broader models that provide an integrated explanation of a large set of findings is similarly complicated. Here, the problem is that the models are built to explain sets of findings that overlap only partially. For instance, some of the introduced memory models, such as the REM model, have only been applied to paradigms from recognition tests, while the temporal clustering and sequencing model has been mostly applied to recall tasks. To compare these comprehensive models, common evaluation metrics are required, aimed at the explanation of a broader domain.

It can be summarized that there are well-established metrics to compare narrow models amongst each other according to their goodness-of-fit while controlling for their complicatedness. However, comparing these models against broader models is ill-defined since there are no agreed upon metrics. Similarly, comparisons between broader models are questionable, as they have their own respective foci. Thus, a designated set of metrics from each respective branch of research is required, according to which the models can be evaluated and compared.

1.3.2 Cognitive architectures as broad models of cognition

Widening the scope from models of individual cognitive phenomena to holistic models of cognition we arrive at the research on cognitive architectures. A cognitive architecture represents an attempt to create an *unified theory of cognition* (Newell, 1994; Vernon, 2014). Allen Newell, the inventor of cognitive architectures, criticised experimental psychology because of the ever-growing pile of unrelated phenomena, making for a fractioned picture of the human mind (Newell, 1973, 1994). To counteract this development, he introduced the idea of cognitive architectures, which originated in the cognitivist tradition, but was equally adopted in the emergent paradigm. A cognitive architecture is conceptualized as a relatively stable system of structures and mechanisms that underly cognition in general. Its specification includes the components of the cognitive system as well as their dynamic relation. Ron Sun, a cognitive scientist working on cognitive architectures, human reasoning, learning and hybrid connectionist-symbolic models, described a cognitive architecture as *"a broadly-scoped domain-generic computational cognitive model, capturing the essential structure and process of the mind, to be used for broad, multi-level, multi-domain analysis of behaviour"* (Sun, 2007, p.160). From Sun's definition, it becomes evident that cognitive architectures are not aimed to explain a specific experimental paradigm but rather to capture cognition as a whole across phenomena. A cognitive architecture contains all major cognitive functionalities and integrates them into a consistent theory of the cognitive system as a whole (Lewandowsky & Oberauer, 2018). According to Allen Newell it is the objectified, complete, executable specification of a theory of cognition, amenable to empirical evaluation (Newell, 1994). David Vernon identifies nine functional capabilities an ideal cognitive architecture has to support (Vernon, 2014, p.73):

1. Recognition and categorization
2. Decision making and choice
3. Perception and situation assessment
4. Prediction and monitoring
5. Problem solving and planning
6. Reasoning and belief maintenance
7. Execution and action
8. Interaction and communication
9. Remembering, reflection, and learning

Additional amendments to the list include multiple representations or different types of memory and learning (Vernon, 2014).

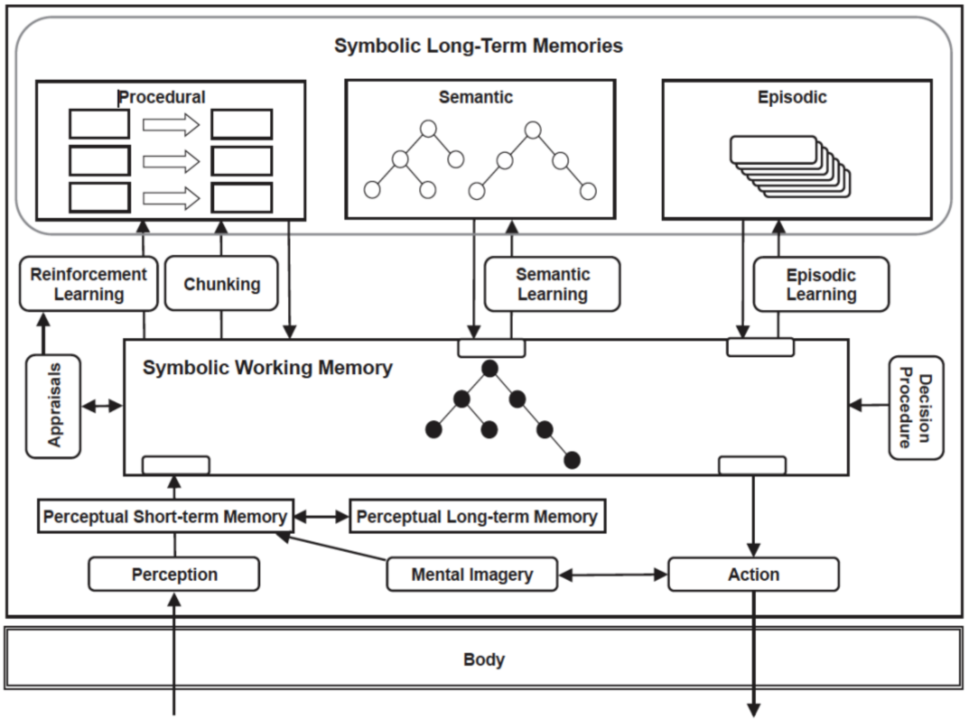
Furthermore, Vernon identifies three different styles of cognitive architectures that can be related to dominating paradigms in cognitive science (Vernon, 2014). A cognitivist perspective, an emergent perspective and hybrid approaches. The cognitivist and the emergent perspective will be briefly introduced together with an example architecture.

In the cognitivist paradigm*, "the focus in a cognitive architecture is on the aspects of cognition that are constant over time and that are independent of the task"* (Vernon, 2014). It is a broadly scoped domain generic computational cognitive model that is neither domain-specific nor task-specific. To enable the architecture it needs to be populated with knowledge, which provides the means to perform a task or behave in a particular way (Vernon, 2014).

A seminal cognitivist cognitive architecture is Soar invented by Allen Newell and continued by his students Paul Rosenbloom and John Laird. Up to this day it is in active development with the latest complete summary being published in 2012 (Laird, 2012). It is focused on processing symbolic knowledge relying on a rule-based production system (Vernon, 2014). Its architecture is comprised of several modules, illustrated in Figure 5. Soar as a cognitive model consists of two essential parts: 1) The *architecture* and 2) Soar's *knowledge* which complements the architecture. Soar is built out of a working memory containing procedural knowledge as IF – THEN rules, semantic knowledge and episodic knowledge (Laird, 2012): To gather new information and behave Soar has a dedicated perception action interface. Soar behaves by searching through problem states via its production cycle it is continuously running through. This cycle consists of 5 steps: In the input step working memory elements are created, which reflect changes in the environment. The second step is the elaboration phase, matching the percepts against the IF parts contained in the procedural knowledge. Hereby, all matching rules are activated, changing the state and in turn activating additional rules. Upon no more rules firing the elaboration phase transitions into the decision cycle, where the options are compared to one another according to numeric preferences. Once the "best" rule is identified a motor command is sent in the application phase, which is executed in the output phase. Soar mostly operates on this cycle, but has additional functionalities like for example *chunking*. *Chunking* enables Soar to deal with situations in which no rule matches. Here Soar creates an *impasse*, indicating a "lack of knowledge". To deal with the impasse a substate is created in which different operators are again evaluated against one another drawing in additional knowledge like for example long-term memory information. After identifying an appropriate operator using the additional information the impasse can be resolved and a new rule is created for future use. In a nutshell, chunking can be thought of as a deductive compositional learning mechanism, where Soar leverages prior knowledge to cope with the unseen circumstances. However, this is only one of Soar's learning mechanisms next to others like for example reinforcement learning. For a thorough introduction see Lehman et al. (1996) or Laird (2012).

Figure 5

The Soar cognitive architecture



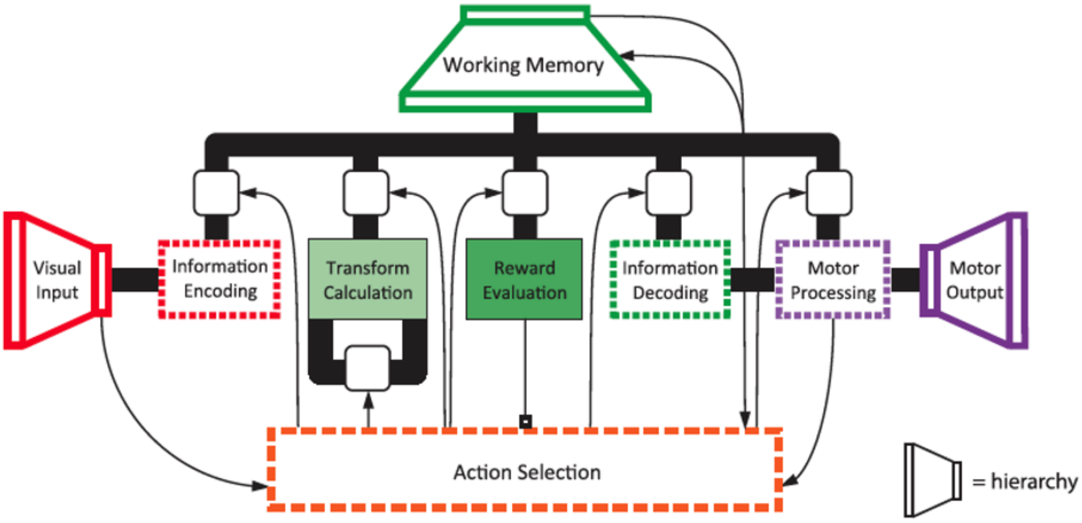
Note. The Soar cognitive architecture. A schematic overview of the Soar cognitive architecture and its individual modules. Taken from Laird (2012, p. 17).

The second style of cognitive architecture is the emergent perspective. While in Soar the architecture is complemented with knowledge, emergent approaches take a different route. Here the framework is not populated with knowledge but rather it is the framework that facilitates development (Vernon, 2014): "An emergent cognitive architecture is essentially equivalent to the phylogenetic configuration of a new-born cognitive agent: the initial state from which it subsequently develops". One example for an emergent cognitive architecture is the Semantic Pointer Architecture Unified Network (SPAUN) developed by Chris Eliasmith

(Eliasmith et al., 2012). SPAUN is a cognitive architecture initially comprised of about 2.5 million neurons, which grew to a total of 6.6 million neurons and draws on insights from neuroanatomy, neurophysiology, and psychology. Each of SPAUNs functional properties map onto areas found in the human brain including visual areas V1 through V4, as well as major memory, motor and cortical areas like for example the dorso- and ventro lateral prefrontal cortex and the orbitofrontal cortex. Unlike Soar it does not rely on a set of production rules but rather on a network of interconnected spiking neurons that form individual modules as illustrated in Figure 6. SPAUN is able to solve eight different tasks based on visual input and implements actions via a physically modelled arm (Eliasmith et al., 2012). A core component are its semantic pointers, which can be understood as state-space representations implemented in spiking neurons. For example, in the case of vision these state-space representations are constructed by applying compression operators onto the visual input abstracting away the raw visual signal and forming a more compressed representation. This compressed information in the task-state representation allows for efficient further processing i.e., selecting and implementing an appropriate action.

Figure 6

Functional architecture of SPAUN



Note. The functional architecture of SPAUN. A high-level depiction of SPAUN with the central features of its' architecture. Taken from Eliasmith et al. (2012).

SPAUN is in active development and the number of tasks SPAUN is able to solve is steadily increasing. The most recent overview of SPAUN as a whole can be found in the book "How to build a brain: A neural architecture for biological cognition" (Eliasmith, 2013).

Next to the symbolic approach as represented here by Soar and the emergent approach as for example SPAUN, there are also *hybrid* cognitive architectures combining ideas of the two. For a review of the research on cognitive architectures see "40 years of cognitive architectures: core cognitive abilities and practical implications" (Kotseruba & Tsotsos, 2020). Here the following chapter will move away from cognitive architectures to different levels cognition can be modelled on. The introduction of individual cognitive models and cognitive architectures were intended to illustrate the different granularities cognitive models can be constructed on. On the one hand, trying to capture isolated small cognitive phenomena. On the other hand, cognitive architectures aiming to capture cognition as a whole.

1.4 Levels to describe, explain, and model cognition

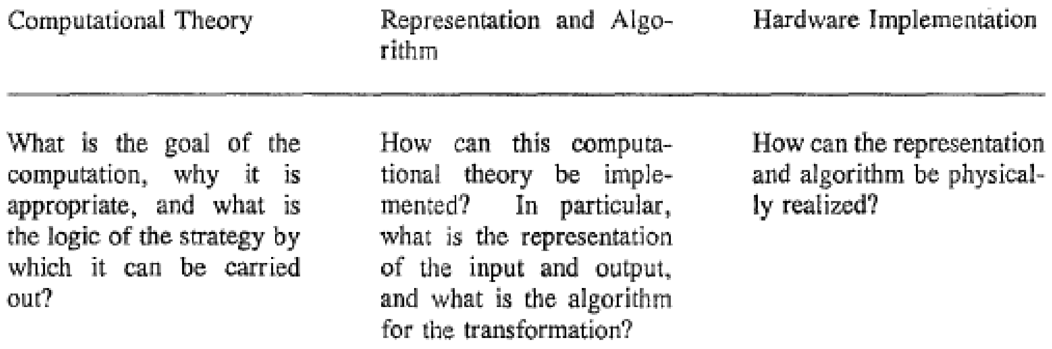
Cognition can be analyzed at different levels of explanation. One distinction which can be made is the dichotomy between bottom-up and top-down approaches (Rauss & Pourtois, 2013). The bottom-up approach tries to understand cognition by studying individual pieces and how their interaction can lead to more complex cognitive phenomena (McClelland et al., 2010). Examples for bottom-up approaches are neural network models or the earlier introduced cognitive architecture SPAUN. Neural network models, as indicated by their name, model the neural system. Therefore, some would argue that they are only tangentially relevant to *cognitive* science (Bechtel, 1994). Cognitive neuroscientists would disagree insisting that the neural systems realizing the cognitive functions are essential to our understanding of cognition as the cognitive functions cannot be understood without understanding their realization in the physical world. One example for models taking a bottom-up approach are models of the brain. Starting out at the neural implementation and the interaction of neurons bottom-up models work their way up and try to answer the question how from these lower level mechanisms higher cognitive functionalities arise. (McClelland et al., 2010). One example are connectionist models of the visual system. Neural network models loosely resembling the human neural system are built to solve object recognition tasks. Hereby the emerging structures within the artificial neural network show some analogies to their biological counterpart and are used as a model for human visual perception. (Güçlü & Gerven, 2015; Yamins & DiCarlo, 2016).

Starting out at the opposing end, classical computational models (CCM) take a top–down approach (Griffiths et al., 2010; Tenenbaum et al., 2011). Classical computational models are best construed as a broad family of process models as introduced in chapter 1, which share a set of properties (Samuels, 2019): They address the higher cognitive functionalities, characterizing their target as computational system of a particular sort (Samuels, 2019). CCMs represent cognitive processes and systems as involving a kind of algorithmically specifiable symbol manipulation. The information about the world is abstracted by perception and represented in symbolic form which can be manipulated. Symbols are representations with semantic properties i. e., they denote or refer to something in the world. Furthermore, they possess formal or syntactic properties and belong to a system of representations akin to a language (Samuels, 2019). Thus, these symbols are characterized by sets of rules, that specify which combinations of symbols are well-formed or grammatical and additionally assign meaning to symbols. Classical computation also presupposes a set of rules in the form of algorithms for how these symbols are to be manipulated. Thus, CCMs are characterizing cognitive processes as algorithmically specifiable processes for computing functions. The in the previous chapter introduced Soar is an example cognitivist architecture manipulating symbolic representations.

One of the seminal analyses addressing the question what levels to describe information processing systems (e.g., cognitive systems) at was advanced by David Marr (1982). Marr proposed a minimum of three distinct, loosely coupled *levels of analysis* which he termed the *computational, the representational and algorithmic,* and the *implementational level,* shown in Figure 7. The computational level specifies the function that the cognitive system is to perform and how it is solved in functional terms. The algorithmic level explicates the used representations and algorithms to implement the processes the mind executes to solve the problem stated at the computational level. The hardware implementation level is focused at the physical instantiation in how the representation and algorithm can be physically realized (Bechtel, 1994; Marr, 1982; Niv & Langdon, 2016).

Figure 7

Marr's levels of analysis



Computational Theory	Representation and Algorithm	Hardware Implementation
What is the goal of the computation, why it is appropriate, and what is the logic of the strategy by which it can be carried out?	How can this computational theory be implemented? In particular, what is the representation of the input and output, and what is the algorithm for the transformation?	How can the representation and algorithm be physically realized?

Note. Marr's three levels any system carrying out an information processing task must be understood at. Taken from Marr (1982, p. 25).

The motivation for Marr's framework stems from Marr's own work in neuroscience, focusing on vision. His research progress stagnated as a result of the failure to discover more mappings from cognitive functions onto individual centers in the brain. The missing progress in combination with his main problem that knowing the responsiveness of particular cells to particular sensory information does not by itself reveal how those neurons contribute to vision resulted in the conclusion that a different level of understanding is required (Peebles & Cooper, 2015).

According to Marr to fully understand a cognitive system, it needs to be addressed at all three levels (Marr, 1982). Exclusively addressing a phenomenon at one level is insufficient. Thus, to address cognition across all levels, the contributing disciplines in cognitive science develop models at the different levels on different scales. On the one hand, in biology and neuroscience models are developed studying how cognitive processes are instantiated in biological substrate, taking a bottom-up approach, trying to show how the interacting pieces generate cognitive functionalities. On the other hand, for example psychologists develop cognitive models often at the computational and the representational level representing a top-down approach which in a next step need to be related to their biological realisation.

Marr's levels of explanation is one taxonomy which has proven to be productive in cognitive science to decompose problems (Hauser et al., 2016; Schulz & Dayan, 2020). There are alternatives like for example Tinbergen's four questions breaking the problem up into evolution, function, mechanism and a developmental aspect (Stephen & Sulikowski, 2019) or Scott Kelso's theory of Coordination Dynamics (Kelso, 1995). The latter was motivated by criticisms of the insufficient focus on the implementational level in Marr's proposition. Kelso's dynamical systems approach proposes an alternative set of three levels emphasizing the relevance of embodiment and situatedness. Proponents of the dynamical systems theory criticize the information processing paradigm and argue for an increased attention on the dynamical interaction with the environment rather than focusing on computations over representations (Colombo & Knauff, 2020). Focusing at the computational level and the functions solving a problem entails a set of assumptions. Functionalism as one doctrine in philosophy of mind focuses on the *function* of cognitive processes and their role in the cognitive system rather than their internal constitution or the substrate they are realized in (Levin, 2018). This entails that a cognitive process specified at the computational level in functional terms and the respective algorithm solving the problem is independent of the implementational level and the physical substrate. “A single mental kind […] can be realized by many distinct physical kinds” (Bickle, 2020): This stance is known as *multiple realizability* and one of the core assumptions in *Functionalism*.

The discussion around functionalism, multiple realizability as well as the right taxonomy to understand cognition is an ongoing debate. This thesis will not go deeper into the discussion about the differences between taxonomies and the respective benefits and shortcomings. Here the focus will be on computational models of cognition and how they can be beneficial for cognitive science research.

Depending on the level of description, the modelling scientist might pick a different type of model that seems the most appropriate for the task at hand. However, there is an ongoing discussion whether for the different levels of analysis different types of models are required or if one model type could suffice and account for cognitive processes across all levels. On the one hand, for cognitive architectures and other models in the cognitivist tradition it remains unclear how these representations and manipulations are realized at the hardware level. On the other hand, connectionist models built upon the analogy to neural processing have yet to demonstrate their capabilities to account for the entire “distribution of latencies in common cognitive tasks” (Farrell & Lewandowsky, 2018, p. 365). It remains to be seen whether a purely cognitivist top-down approach or exclusively bottom-up models in the connectionist tradition suffice to account for all phenomena across all levels (Bechtel, 1994). Although neural networks in the connectionist tradition appear very different from “classical” computational models in the cognitivist tradition, the two are not mutually exclusive (Rescorla, 2020). There are also hybrid approaches working on the question how the computations specified at the higher levels can be realised in connectionist neural network models, combining the two modelling approaches (Marcus, 2001). For example, research on neural networks has demonstrated that classical models can be implemented in neural networks (Bechtel & Abrahamsen, 1991; Rescorla, 2020). Also, in the cognitive architecture research there are hybrid approaches like Paul Smolensky's Integrated Connectionist/Symbolic Cognitive Architecture (ICS) (Smolensky, 1990).

Furthermore, in recent years the idea of a joint effort by bottom-up and top-down approaches received increased attention embodied in the newly emerging discipline computational cognitive neuroscience (Naselaris et al., 2018). Here the idea is to reconcile the varying approaches rather than thinking of them at opposing ends competing against one another for explanatory primacy. The idea is to build functional cognitive models guided and informed by insights from neuroscience, whereas the two are in a complementary relationship mutually constraining each other (Kriegeskorte & Douglas, 2018).

2 Bayesian methods in cognitive science

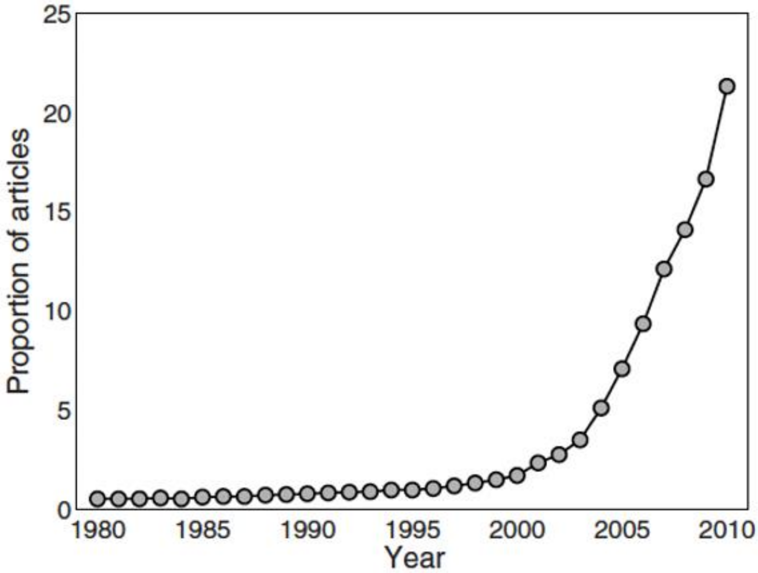
After this introduction into different modelling approaches including the distinction between statistical, descriptive models and cognitive models as well as the different levels of explanation that scientists model cognition on, we will now focus on *Bayesian methods* for cognitive modelling.

2.1 The motivation for the Bayesian approach

The Bayes' principle developed by Thomas Bayes' was already published in the 18th century (Bayes & Price, 1763). However, only recently it was picked up in cognitive science research and received increasing attention over the past years. Two examples are 1) the increasing amount of papers published matching the search for "bayes" shown in Figure 8 in the timespan form 1980 to 2010, adapted from Wagenmakers & Lee (2014).

Figure 8

Popularity of Bayesian modelling



Note. Google scholar search showing the articles matching the search "bayes OR Bayesian – author: bayes" from 1980 to 2010. Adapted from Lee & Wagenmakers (2014, p.8).

The figure shows that from the year ~2000 onwards the number of articles matching the search "bayes OR Bayesian – author: bayes" exponentially increased. This large interest in

Bayesian models is ever ongoing. At the moment of this thesis in December 2021 the interest seems to not have plummeted. A Google scholar search for “bayesian modeling” returns approximately 152000 articles many of which coming from the field of cognitive science.

Going in line with this trend is 2) the substantial addition of chapters dedicated to Bayesian methods in one of the seminal textbooks on computational modelling in cognitive science. In 2011, the first edition of “Computational Modeling in Cognition” was published by Lewandowsky & Farrell (2011). Up to that point the increased attention in Bayesian methods did not yet find its way into Lewandowsky & Farrell's textbook. However, this changed with the release of the second edition in 2018. One of the major changes from the first to the second edition was the addition of four specifically dedicated chapters on Bayesian methods

These examples immediately raise the question what Bayesian methods offer to cognitive modelling: Proponents of Bayesian statistical methods argue that Bayesian methods provide a complete and coherent framework for the challenge of relating scientific models to data (Jaynes, 2003; Lee, 2018). The defining feature of the Bayesian statistical approach is its use of probability distributions to represent uncertainty (Lindley, 1972). *Prior probabilities* over models and their parameters are transformed to posterior probabilities using Bayes’ rule, informed by the evidence provided by the data. This is the Bayesian principle in a nutshell. We will explore it more thoroughly in the following chapters. Beforehand, another distinction in Bayesian modelling needs to be pointed out. There are different types of models constructed based upon Bayes’ rule. Thus, the chapter will start out describing the different models built in cognitive science based on Bayes’ rule following Lee’s article “Bayesian methods in Cognitive Modelling” (Lee, 2011, 2018). These different use cases of Bayes’ theorem can be puzzling since different concepts are referred to by the same name. Unravelling these different concepts and resolving my personal confusion that some might encounter themselves was one of the motivations for this thesis.

2.2 Types of Bayesian methods in cognitive science

The first application of Bayesian methods in cognitive science are *data analysis methods*. In cognitive science, statistical methods are a cornerstone of empirical research to take the step from individual observations in an experiment to empirical generalizations. Most of the statistical methods taught at universities are in the *frequentist* tradition based on the generalized linear model. Nevertheless, some statisticians vouch for a reform of the curricula and argue that classical statistical methods have serious conceptual and practical limitations (Lee & Wagenmakers, 2014; Wagenmakers, 2007). Bayesian statistical methods could potentially replace classical frequentist concepts as *t*-tests, *F*-tests, *p*-values. A replacement could be accomplished by considering the same statistical model as for the classical test, but applying Bayesian methods for the inference (Kruschke, 2010). There are proponents of the classical frequentist approach as well as proponents of the Bayesian statistical methods. For a more thorough discussion of the different statistical approaches and an introduction to the two see the book “All of statistics” by Larry Wasserman (2013).

The second application of Bayesian methods in cognitive science are *Bayesian models of the mind*. Bayesian statistics provides a solution to the problem of making inferences about structured hypotheses based on noisy, sparse data (Lee, 2018). Thereby, it provides a useful and potentially compelling metaphor for the mind. To understand the mind as a mechanism solving problems according to Bayesian principles has proven to be a productive metaphor resulting in many models of cognition, often pitched at the computational level of Marr’s levels of analysis (Chater et al., 2006; Marr, 1982). Many different cognitive phenomena have been addressed by Bayesian models of the mind including vision (Kersten & Yuille, 2003), language (Culbertson & Smolensky, 2012; Narayanan et al., 1998), decision making (Vul et al., 2014), and development (Gopnik & Tenenbaum, 2007). The discussion whether this “Bayes’ in the head” argument is adequate is controversial and yet to be resolved (Jones & Love, 2011; Tauber et al., 2017).

From these individual applications of Bayes’ rule to cognitive phenomena there is an attempt to expand the concept to the entire brain aimed at developing a “unified principle of cognition”. Similar in spirit to Newell who argued in favor of a unified theory of cognition, proponents of the “Bayesian brain hypothesis” or predictive processing account propose a framework in which *all* of cognition is governed by a *prediction error minimization* process. This process relies on Bayes’ rule to combine previous knowledge with new incoming information (Clark, 2015; Hohwy, 2013; Knill & Pouget, 2004). This idea of prediction error minimization as the core principle of cognition was applied to many different areas. A seminal example is sensory perception, for example vision (Hohwy et al., 2008). Here, the perceived percept is explained as a combination of previous knowledge i.e., knowledge about the context one is embedded in and newly incoming information i.e., the visual input. Both information streams are integrated to form the posterior belief which is the resulting percept. Next to vision, the Bayesian brain hypothesis has been applied to many areas such as research on emotions (Seth, 2013) or clinical psychology (Horga et al., 2014; Palmer et al., 2015). However, the research field is still in its early stage, and it remains to be seen whether the assumptions and predictions can be empirically supported. At the present moment, many questions remain unresolved, for example, the question of how the prediction error minimization is implemented in the neural substrate. A thorough introduction accompanied by various statements of active researchers in the field can be found in “Whatever next? Predictive brains, situated agents and the future of cognitive science” (Clark, 2013). Furthermore, the book “The Philosophy and Science of Predictive Processing” explores current ideas and ongoing work in the predictive processing framework (Mendonça et al., 2020).

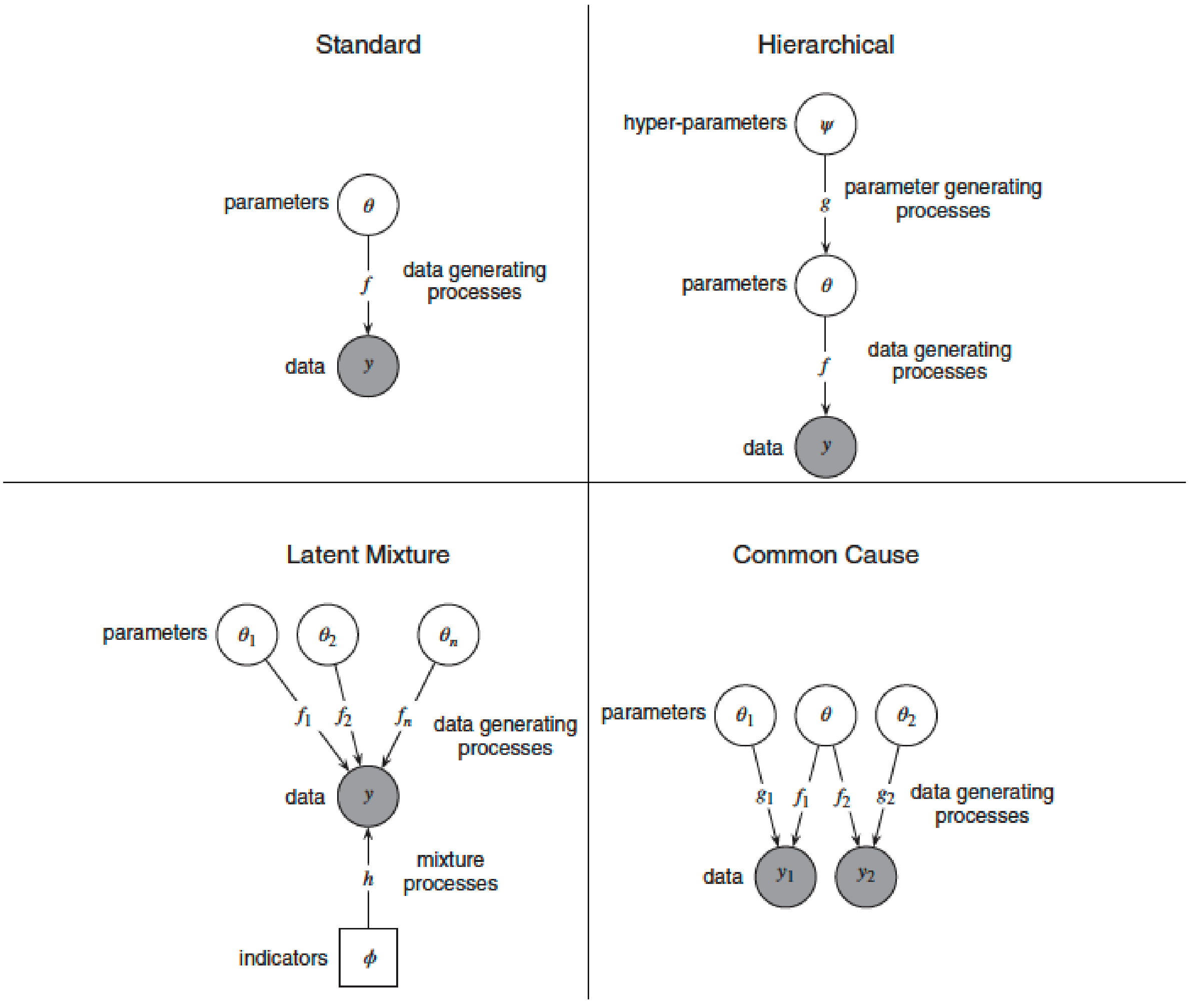
The third application, the one which is the focus of the remainder of this thesis, is Bayesian techniques for assessing cognitive models. In this case, the Bayes’ rule is used to estimate parameters and assess cognitive models in the light of collected data. Importantly, the application of Bayesian techniques for assessing cognitive models does not necessarily claim that the mind or our models implement Bayesian processes. Despite the shared name these two kinds of Bayesian models are conceptually different.

2.3 Bayesian cognitive models

Generally, the value of Bayesian methods for cognitive modelling stems from two complementing strengths (Lee, 2018): On the one hand, Bayesian methods offer a principled foundation for statistical inference. On the other hand, Bayesian methods offer the creative freedom and modelling flexibility to develop and test a wide range of cognitive models. Figure 9 visualizes different classes of cognitive models. All of these can be analyzed by use of the Bayesian parameter estimation techniques which will be introduced later.

Figure 9

Bayesian cognitive model examples



Note. The figure gives an overview of the different cognitive models made possible by the Bayesian approach. "*The standard model defines a process f controlled by parameters θ for generating behavioural data y. A hierarchical model structure extends the standard model by including a process g controlled by parameters ψ that generates the original parameters θ. The latent mixture structure allows for different data-generating processes f_1 to f_n controlled by different parameters θ_1 to θ_n to combine to generate the data, according to some mixing process h controlled by parameters ϕ. The common cause model structure allows for different data y_1 and y_2 to be in part generated by the same parameters θ.*" A more detailed explanation is given in the text. Figure adapted from Lee (2018, p. 5).

The top left panel in Figure 9 shows the structure of a standard cognitive model, involving cognitive variables θ, controlling cognitive processes f that generate behavior y. In this illustration the dark grey circle indicates a manifest variable while the white circle indicates a latent variable. In this case, the cognitive model takes the form $y = f(\theta)$, which is the form most cognitive models can be conceived as. Often, the function f is very complicated, potentially involving multiple processes. Nonetheless, it constitutes a single mapping from parameters to data. Based on this mapping, Bayes' theorem allows for prior knowledge about parameters to be updated to posterior knowledge using the information provided by data.

The appeal of the Bayesian appeal stems from its transparency regarding the uncertainty of parameter estimates as well as from its inherent flexibility (Lee, 2018, p.39): The top right panel in Figure 9 shows a *hierarchical* structure. It is based on the assumption that the parameter θ itself is generated by a parameter generating process g based on hyperparameters ψ. This takes the theorizing one step further. For instance, let $f(\theta)$ be the generating process of the behavioural performance in a memory task. Then $\theta=g(\psi)$ could model and represent the hyperparameters and processes leading up to a certain memory capacity, like the underlying neural processes.

The bottom left panel shows a *latent mixture model*. Here rather than one parameter and one process, multiple processes f_1 to f_n based on multiple parameters θ_1 to θ_n generate the observable behaviour y. How these processes interact and are combined is controlled by a mixing process h that itself is indexed by the parameter ϕ. These models are especially suited for modelling scenarios in which multiple participants potentially apply different strategies to solve a task. Here, the index parameter indicates which approach was chosen and allows to model these processes separately.

The last introduced expansion from the standard model is the *common cause model*, shown in the bottom right panel. In this case, the assumption is that some psychological variables influence *multiple* sorts of cognitive capabilities: Different datasets y_1 and y_2 potentially coming from different experiments and stimuli both depend on the same psychological variables θ_i. This leaves the option that there are potential task-specific parameters and processes exclusive to y_1 or y_2 but allows to model some common cause across the two tasks.

Each introduced model structure i.e., hierarchical, latent-mixture and common cause models build upon the standard structure allowing the formalization of more elaborate accounts of cognition. However, none of these models are inherently Bayesian. The Bayesian methods applied to these structures make them “Bayesian” models. These methods can be applied to all model structures making it a very versatile, powerful modelling tool. This enables the modeler to build detailed, speculative models of cognition and in a next step to evaluate them against data. Quantitative fit is one-dimension along which models can be evaluated. It provides an estimate how well a model fits the data and enables to compare models against each other. However, it is important to note that although a good quantitative fit of a model is a strong argument in favor of a it, it is only one dimension along which models can and should be assessed. The importance of quantitative fit and other dimensions to evaluate models are introduced in chapter 1.3.

2.4 Parameter estimation techniques

How does Bayesian inference allow us to relate models to data to eventually evaluate and compare our models against each other? Models themselves have a fixed structure. For example, a linear model has a fixed structure fully determined by its slope and y-axis intercept. The slope and intercept are given by the parameters which are obtained by fitting the model to the data. Farrell and Lewandowsky (2018, p. 38) describe parameters as the “tuning knobs” that fine-tune the behaviour of a model once its architecture is defined. The analogy is a radio with a fixed architecture corresponding to the model and its “knobs” determining volume and the station, without altering the overall architecture. The process of finding the right setting for the knobs on the radio is the equivalent of fitting our model to the data by finding ideal values for the free parameters. While for the radio it is the search for the ideal sound, for a cognitive model it is the estimation of the ideal parameters that align the predictions of the model with the observed data to the best extent possible.

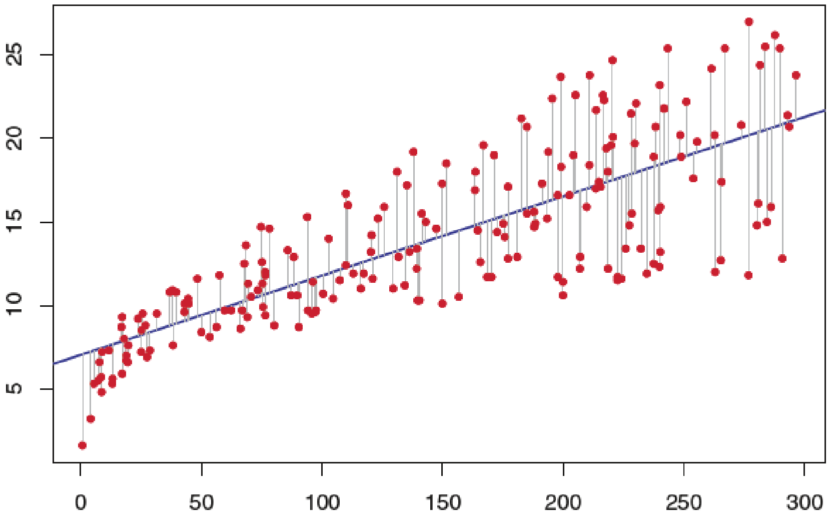
The estimation of the ideal parameters can be accomplished using different approaches. At first two non-Bayesian methods for parameter estimation will be introduced followed by the Bayesian approach which will be the centre of focus.

Because the overall behaviour of the model depends on its parameters, it is crucial to carefully fit its parameters. When fitting models to data it is necessary to pay attention and avoid overfitting (James et al., 2013). Overfitting describes a model that fits the collected data well but does not generalize well to new unseen data. This problem is especially prominent in more complex non-linear models with many parameters that can be specifically tuned to a given dataset. There are multiple approaches to cope with overfitting one of the most prominent being cross-validation. In cross-validation a model is repeatedly trained on parts of the data and tested on a hold-out set. This estimates the model’s ability to generalize as the model is evaluated on data that was not included in the parameter estimation. To evaluate and compare a model the model must be fit to the data well, while controlling for a potential overfit.

The first approach to find the best parameters of a model is the *minimization* of a *discrepancy function*. The discrepancy function describes the deviance between predictions and data. The minimization requires a continuous discrepancy function that condenses the discrepancy between the predictions and the data to a single number. Then the discrepancy function can be minimized by iteratively adjusting the parameters. One example for a discrepancy function, which is also referred to as error function, is the *root mean squared error deviation* (RMSD). It summarises the average of the squared deviations between the data and the predictions, and is illustrated in Figure 10. For an introduction to other discrepancy functions see Chechile (1998, 1999). The error function can be visualized as a surface in the case of 2 parameters like the intercept and slope from Figure 10. Figure 11 shows the error surface with the RMSD on the y-axis and the intercept and slope on the x- and z-axis.

Figure 10

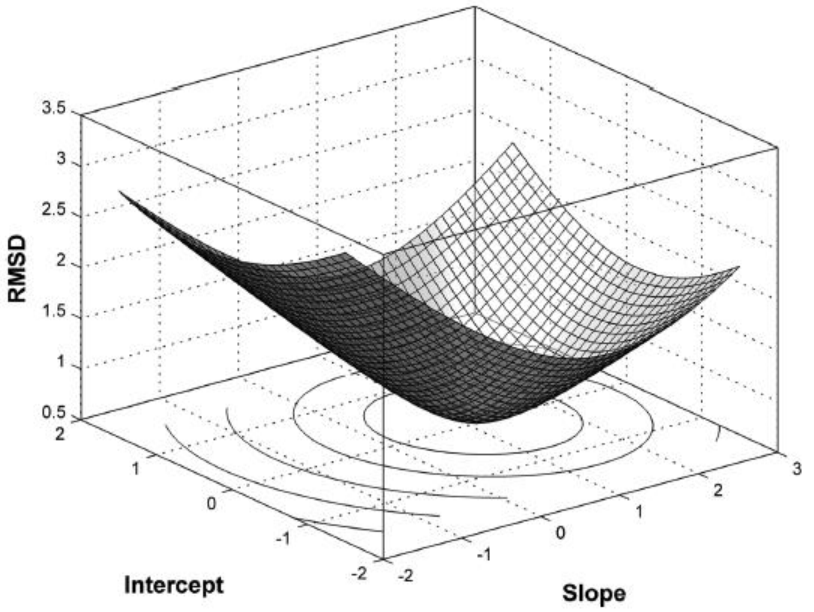
Model fitting by minimizing the residuals



Note. The straight line represents a linear model fitted to the data. The red dots indicate individual measurements. The slope and intercept are obtained by minimizing the root mean squared error deviation. Taken from James et al. (2013, p.63).

Figure 11

Error surface visualization to obtain ideal parameters



Note. Figure 11 displays the error surface for a linear model. The slope and intercept axes correspond to the free parameters in the linear model, while the y axis displays the root mean squared error deviation. The minimum of the surface corresponds to the ideal parameters minimizing the error function. Taken from Farrell & Lewandowsky (2018, p.51).

The ideal parameters are those that correspond to the minimum of the plotted surface. Often, for psychological models there is no direct algebraic solution as it is the case for a linear regression for example. Thus, iterative methods searching for the best parameters are applied. One approach to find this ideal combination of parameters is to step through the discretized parameter space. The parameter space is searched at specific increments and cutoff at an upper and lower bound. After searching the space, the parameter combination which yields the lowest RMSD is chosen. This approach is referred to as grid-search. However, with more than two parameters the search space quickly grows. Overall, the search space grows exponentially with adding additional parameters, which is referred to as the "curse of dimensionality". Thus, for more complex problems grid-search becomes impractical and other approaches are required to estimate model parameters. (James et al., 2013, p.242). Other approaches search the parameter space more systematically exploiting the gradient in the error surface. These approaches include the *Simplex* algorithm, *Simulated Annealing,* and *Gradient Descent*. All of them have their individual strengths and shortcomings. For a more detailed introduction to their workings see Bishop (2006) and Farrell & Lewandowsky (2018, p.46ff).

The second prominent approach to estimate parameters is *Maximum Likelihood maximisation*. Here, rather than finding the parameters minimizing the discrepancy between data and predictions, the starting question is: Given the data y, what is the likelihood of the values in the parameter vector θ? So, to find the parameter values which are unknown, we aim at maximising the likelihood of observing the data we already collected. In other words, given the built model and the already collected data the task is to estimate the parameters that explain the data the best (Farrell & Lewandowsky, 2018). A thorough introduction to the concept of likelihood and maximum likelihood methods can be found in Farrell & Lewandowsky (2018, p.72ff) or Held & Bové (2014). In short, these methods can be understood as solving the parameter estimation in a similar manner to the minimisation of the discrepancy function. The latter searches the parameters minimizing the discrepancy between prediction and the data. Maximum Likelihood tries to find the maximum of the likelihood, thus the peak of the likelihood function. However, the search through the parameter space can again be accomplished by applying methods like Simplex or Simulated Annealing in a slightly adjusted version to account for searching a peak rather than the *tale* of a function.

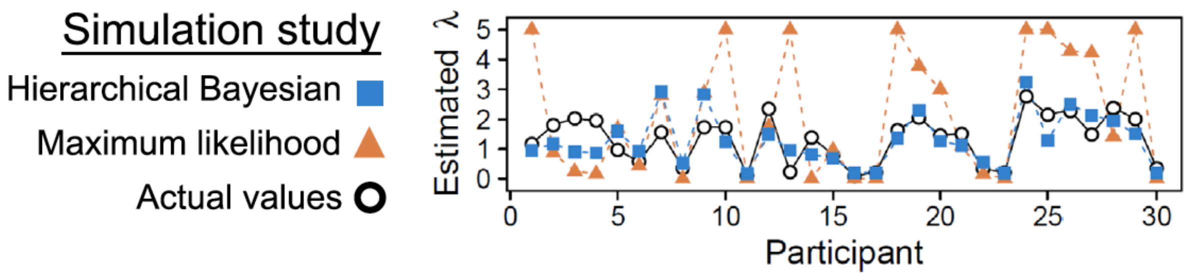
Now, we will finally start looking at Bayesian modelling in cognitive science, beginning with a short introduction to Bayes' theorem itself and then moving on to Bayesian parameter estimation techniques. In the last section, the concept of likelihood was briefly introduced as the means of identifying the most likely value of a parameter in the light of the observed data. However, we have to be cautious. Likelihoods permit *relative* comparisons amongst different parameter values. Thus, they are suitable to maximise likelihoods in order to obtain parameter estimates – but they are not suited for estimating absolute probabilities (Farrell & Lewandowsky, 2018). The resulting problems become evident in a simple example from Farrell & Lewandowsky (2018). Suppose after estimating a parameter given a dataset from a memory experiment, which investigates how many words can be stored in working memory, the estimated parameter value is four. Without going into depth of the design, knowing that there always is some measurement error associated with our measurement, this punctuated estimate tells us relatively little. What would be more informative is the *range* of the "true" parameter value, in the form of a *probability distribution*. This cannot be achieved by maximum likelihood estimates but can be solved using Bayesian parameter estimation. The Bayesian approach generates a posterior distribution, which can be analysed and inherently represents the certainty about a parameter estimate, while the maximum likelihood approach generates a point estimate.

A simulation study conducted by Ahn et al. (2017) further revealed that the Bayesian approach outperforms maximum likelihood approaches on small datasets as shown in

Figure 12. The Bayesian approach generates more precise parameter estimates, making a compelling argument in favor of Bayesian parameter estimation techniques.

Figure 12

Comparison of Maximum Likelihood and Bayesian parameter estimation performance



Note. The performance of Maximum likelihood parameter estimation compared to Bayesian Hierarchical approaches. In the study by Ahn et al (2017) Hierarchical Bayesian methods outperformed the Maximum likelihood approach. Taken from Ahn et al (2017).

2.5 Bayes' theorem

Bayes' theorem is grounded in probability theory. Probability theory itself starts off with three fundamental axioms (Farrell & Lewandowsky, 2018):

1) Probabilities of events must lie between 0 and 1.
2) The probabilities of all possible outcomes must sum exactly to 1.
3) In the case of *mutually exclusive events*, the probability of any of the events occurring is equal to the sum of their individual probabilities.

From these assumptions, other useful properties of probabilities follow. One of them is the notion of a *joint probability*, denoted *P*(a, b), which gives the probability that both a and b occur. To compute the *joint probability* *P*(a, b), the individual probabilities have to be multiplied e.g., 2.1.

$$P(\mathrm{a}, \mathrm{b}) = P(\mathrm{a}) \times P(\mathrm{b}) \qquad 2.1$$

However, this is assuming that the two are *independent*. If we know that there is a conditional relationship a|b between a and b, the *joint* probability is given by

$$P(\mathrm{a, b}) = P(\mathrm{a|b}) \text{ x } P(\mathrm{b}). \qquad 2.2$$

The conditional probability describes the probability of observing an event *a* given that we observed event *b,* denoted as *P*(a|b).

At this point we have all necessary pieces to derive Bayes' theorem. A simple example will help to understand how we arrive at Bayes' theorem, taken from Farrell & Lewandowsky (2018). Assuming the joint probability P(a, b) describes the probability of a wet street and a broken water main. This can be calculated using conditional probability, by multiplying the probability that a water main has burst with the probability that the road will be wet, given a burst pipe P(a|b). This takes the same form as in equation 2.2. We now can reverse it for the other direction of conditionality e.g., 2.3.

$$P(\mathrm{a, b}) = P(\mathrm{b|a}) \text{ x } P(\mathrm{a}) \qquad 2.3$$

This would correspond to the probability of a wet street and a broken water main, just as before, which can be calculated by multiplying the probability that the road will be wet with the probability that the main has burst, given it is wet. Because both equations come to the same result, we can combine them as

$$P(\mathrm{b|a}) \text{ x } P(\mathrm{a}) = P(\mathrm{a|b}) \text{ x } P(\mathrm{b}). \qquad 2.4$$

If we now divide both sides by P(a), one conditional probability is expressed as a function of the other:

$$P(\mathrm{b|a}) = \frac{P(\mathrm{a}|b) \text{ x } P(\mathrm{b})}{P(a)}. \qquad 2.5$$

At this point, we already arrived at Bayes' theorem. Equation 2.5 is Bayes' theorem, which now can be rewritten as

$$P(\theta|y) = \frac{P(y|\theta) \text{x } P(\theta)}{P(y)}. \qquad 2.6$$

Now, we can calculate the probability of our parameters θ given our data *y*. This goes beyond what was possible using the likelihood in L(θ|*y*), which allowed for relative comparisons of parameters. It becomes possible to calculate actual probabilities. That is, we can give a *probabilistic interpretation*. For our memory example from earlier, for instance, we could draw conclusions like: “There is a 95% probability that the capacity for the working memory is between 2 and 4”.

First, we will introduce some commonly applied terminology describing the different components in Bayes’ theorem (Equation 2.7) and then turn to the question of how we can calculate the probabilities we are interested in, following the explanations from Farrell & Lewandowsky (2018).

$$\underbrace{P(\boldsymbol{\theta}|y)}_{posterior} = \underbrace{(P(y|\boldsymbol{\theta})}_{likelihood} \times \underbrace{P(\boldsymbol{\theta}))}_{prior} \; / \; \underbrace{P(y)}_{evidence} \qquad 2.7$$

Priors represent our knowledge of our model’s parameters before we collect the data in an experiment. They come embodied in a distribution, which can be discrete or continuous, depending on the nature of the parameter. We can either build upon already existing knowledge or alternatively use weekly or uninformative priors. The latter impose little or nothing onto the posterior, which is then fully dependent on the collected data. In fact, a Bayesian parameter estimation with uninformative priors can result in the same parameter estimation as a Maximum Likelihood estimation (Robert, 2007, p.166). Whereas the mean of the posterior corresponds to the point estimate generated by the Maximum likelihood estimation.

Likelihood: After collecting our data y, we can examine the probability of having obtained a particular outcome in the light of the prior values of the parameters θ. This presupposes that the data is already collected and requires a prior distribution of the parameters.

Evidence refers to the overall probability of the data, irrespective of the parameter values. It acts as a normalization factor, ensuring that the probability is in the range from 0 to 1.

The Posterior probability of the parameters θ results from applying Bayes’ theorem. Like the prior, the posterior is a probability distribution. It is the transformed prior distribution, incorporating the evidence provided by the data. This distribution can be used to provide valuable insights. For instance, the *mode* can be interpreted as the most probable parameter value. Moreover, next to a single point estimate, the posterior distribution directly allows to assess the *certainty* of our parameter value by analysing the width of the obtained distribution.

For our memory example the mode of the distribution could correspond to the best estimate for working memory capacity, accompanied by a certainty estimation in form of the width of the distribution.

To compute the posterior, we need to transform the evidence *P*(y), also referred to as *marginal likelihood*. As just stated, the evidence is calculated irrespective of the parameter values and can be computed by summing across all possible parameter values, as shown in 2.8

$$P(y) = \sum_{\theta} P(y, \theta). \qquad 2.8$$

Now the joint probability can be transformed to their equivalent conditional probability given by

$$P(y) = \sum_{\theta} P(y, \theta) = \sum_{\theta} P(y, \theta) \text{ x } P(\theta). \qquad 2.9$$

If we now plug this transformed version of P(y) into Bayes' theorem, we end up with the most useful version of Bayes' theorem e.g., 2.10.

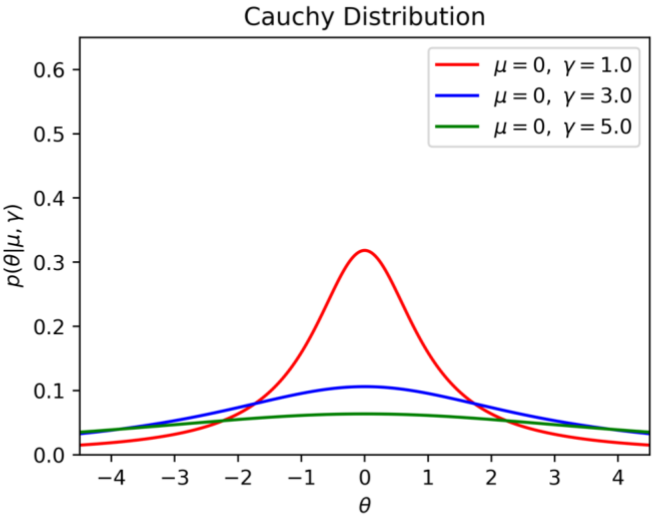
$$P(\theta|y) = \frac{P(y|\theta) \text{ x } P(\theta)}{\sum_{\theta} P(y,\theta) \text{ x } P(\theta)} \qquad 2.10$$

Equation 1.10 states that "the posterior distribution is given by the ratio of the particular outcome that was observed in an experiment given our prior knowledge of the parameters, and the space of *all* possible outcomes that could have been observed in light of our prior knowledge" (Farrell & Lewandowsky, 2018, p.130).

To now compute the posterior, we have to specify the individual parts and apply Bayes' theorem. The prior is defined either by previous research, theories or logical assumptions, e.g., working memory capacity cannot take negative values. If there is no previous knowledge, only weakly informative priors can be chosen. These take the form of flat distributions like for example a Cauchy distribution with a large standard deviation as shown in Figure 13.

Figure 13

Cauchy distribution as prior at varying standard deviations



Note. Figure 13 shows a Cauchy distribution which is a common choice as prior. With increasing standard deviation (γ), the prior is wider and thus has less impact on the posterior distribution. The mean value (μ) of all distributions is the same.

A common example to illustrate the individual components in Bayes' theorem is a series of coin tosses (Farrell & Lewandowsky, 2018, p.130ff). The likelihood for a coin toss is given by a Bernoulli distribution. After having collected a series of coin toss outcomes and having specified a prior, the likelihood can be directly computed. The last remaining component is the *evidence* or *marginal likelihood* given by the overall probability of obtaining that data across all possible parameter values. This last part turns out to be the most difficult. The problem is that it is frequently intractable because of the integral over *all* possible parameter values, which cannot be solved analytically. In the case of a simple coin flip example, the posterior can be obtained without explicitly writing down the marginal likelihood, thus circumventing the problem. The complete example and its analytic solution based on a trick, leveraging the beta distribution as a prior, can be found in the chapter on "Analytic methods for obtaining posteriors" by Farrell & Lewandowsky (2018, p.130ff).

Here, we will resort to the other alternative approaches when the posterior cannot be calculated directly: Thus, instead of analytically solving the equation to obtain the prior, we *estimate* the posterior distribution by simulation. One set of methods the following chapter focuses on are referred to as *Monte Carlo* methods (Farrell & Lewandowsky, 2018, p.146ff).

2.6 Bayesian parameter estimation and Monte Carlo methods

The general idea is straightforward: When the posterior distribution is unknown and cannot be calculated analytically, it is replaced by a large number of samples whose properties mirror those of the posterior distribution. This sampled approximation can be analysed to answer questions about model parameters. In order to be able to sample from the posterior distribution, the prior distribution is required, which is previously defined by formulating assumptions about it. Additionally, the likelihood is needed, which we have knowledge about based on our collected data and given the assumed parameters. These components suffice to compute samples for any particular parameter, here referred to as θ, in the light of the observed data that are at least proportional to the posterior density (Farrell & Lewandowsky, 2018, p.146). The samples are proportional because the marginal likelihood is a constant with respect to θ and hence can be ignored if we limit our interest to relative comparisons. Thus, Bayes' rule is often summarised as

$$P(\theta|y) \propto P(y|\theta) \text{ x } P(\theta), \qquad 2.11$$

where $\propto$ stands for "proportional to". Thus, if we have access to the information on the right-hand side, we are able to sample from the posterior distribution using so-called Markov Chain Monte Carlo (MCMC) simulations. There are three parts to it, which we explain following Farrell & Lewandowsky (2018) and van Ravenzwaaij et al. (2018):

1) "Monte Carlo simulations" refer to a random number generation process from a proposal distribution. For example, we draw samples for θ_t from a normal distribution as *proposal distribution* with mean 0.5 and an arbitrary variance σ, shown in equation 2.12

$$\theta_t \sim N(0{,}5, \sigma). \qquad 2.12$$

   If we sample many times from this proposal distribution and plot the drawn values of θ in a histogram referred to as density plot, the resulting distribution will approximate the proposal distribution given sufficiently many drawn samples.

2) A *Markov Chain* is a sequence of numbers where each number is only dependent on the directly preceding one. For our example, this means that we sample θ_t from a normal distribution, where the mean of the distribution on each draw is the respective preceding value θ_{t-1}, as shown in 2.13

$$\theta_t \sim N(\theta_{t-1}, \sigma). \qquad 2.13$$

If we now sample, always updating the mean of our proposal distribution, the produced histogram in the density plot does no longer approximate our proposal distribution but *wanders*, showing a so-called *random walk* behaviour.

3) The third part is the *Metropolis-Hastings algorithm* that determines which value of θ to accept and add to our histogram, and which to reject. As a first step, an initial plausible value for theta is chosen and in a next step noise is added to the proposed initial value of theta resulting in θ_{t-1} and θ_t. Now, the posterior probability is calculated for both θ_t and θ_{t-1} by multiplying the prior with the likelihoods. Subsequently, the ratio between the posterior probability of θ_t and the posterior probability of θ_{t-1} is calculated. If the posterior probability of θ_t is larger, the ratio is always greater than 1 and the value θ_t is accepted and will be added as a sample to our distribution. If the ratio is smaller than 1, the proposed value for theta is not immediately rejected but compared to an acceptance probability. Now, a random number u is drawn from a Uniform distribution

$$\mathrm{u} \sim Uniform(0, 1). \qquad 2.14$$

The posterior probability ratio is compared to the drawn number u. If the ratio is larger than u, θ_t will be accepted and added to our drawn samples. At this point, the process starts over again. The just accepted θ_t becomes $\theta_{t-1}$and a new value θ_t is sampled to then calculate the posterior probabilities as done before. However, if u is larger than the posterior probability ratio, the proposed value for theta is rejected. The old value for theta is reused and a new proposed value is sampled to then calculate the posterior probabilities over again.

This procedure is repeated many times, rejecting, and accepting proposed values for theta. The accepted values for theta in the end form the posterior distribution which can be analysed. This distribution contains all available information and can generate insights about the parameters of interest. Additionally, next to the mode of the distribution, the width directly enables insight about the uncertainty of the estimate. If the distribution is narrow around the mode, the estimate is precise. If the distribution is wide, other values of theta are plausible and the interpretation should take this uncertainty into account.

The just explained Metropolis Hasting Algorithm is a special form of MCMC simulation and dates back several decades. Over recent years, further developments in the field of MCMC techniques have built upon the Metropolis Hastings algorithm and further refined it.

For the case study in the following chapter, which will show the sampling applied to an example, the software Stan was used (Carpenter et al., 2017): Stan implements a special form of MCMC method named Hamiltonian Monte Carlo (HMC). The main difference between the Metropolis Hastings algorithm and HMC is the proposal distribution. The Metropolis Hastings algorithm uses a symmetrical proposal distribution. HMC, on the other hand, uses a proposal distribution that changes depending on the current position. HMC figures out the direction in which the posterior distribution increases, referred to as its gradient, and warps the proposal distribution toward the gradient. Thereby, drawing values for θ with high posterior probabilities, resulting in more accepted values. For a more thorough introduction to HMC and Stan see "Stan: A probabilistic programming language" and „Doing Bayesian data analysis: A tutorial with R, JAGS, and Stan" (Carpenter et al., 2017; J. Kruschke, 2014).

We will now turn to the case study and apply the just introduced sampling techniques to an example on intertemporal decision making. MCMC simulations will be used to evaluate three different models against data and to estimate the free parameters specified in the models.

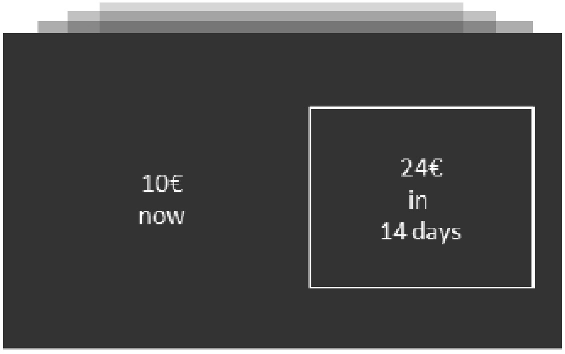
3 Bayesian Parameter Estimation in Practice

3.1 Introduction – Delay Discounting Example:

Intertemporal decision-making generally describes decisions involving consequences at different points in time. The primary theoretical paradigm that has been used to explain intertemporal choices is the theory of delay discounting, which assumes that the greater the delay in receiving the rewards, the more the rewards are discounted (Frederick et al., 2002). Our example involves an intertemporal decision-making task with repeated decisions between two monetary reward options called the Money Earlier or Later (MEL) task. The options to choose from are either a small immediate monetary reward or a larger delayed reward. Figure 14 displays an example trial where the decision is between a 10€ immediate reward and a 24€ reward with a 14-day delay.

Figure 14

Example Trial in Money Earlier or Later Task



Note. Figure 14 shows an example trial in a Money Earlier or Later task.

The data for the model evaluation was provided by Antonius Wiehler and Lei Zhang from the University Clinic Hamburg Eppendorf. It is a simulated dataset based on an experiment conducted by Wiehler et al. (2017). Overall, the dataset contains 20 subjects and 120 trials per subject. The choices were always binary with an immediate reward of 10€ versus a varying larger and delayed reward. The delay varied between 1 day and 170 days, the delayed reward ranged between 10.5€ and 49.8€.

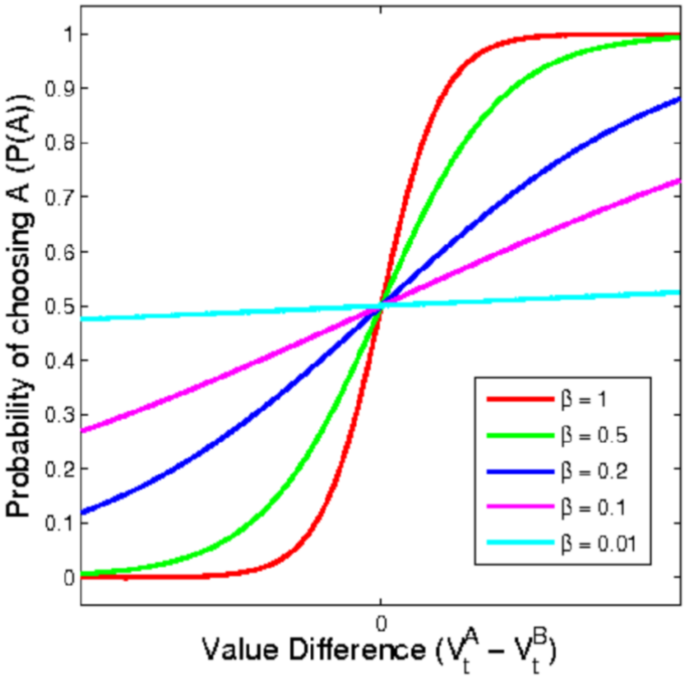
The literature on subjective value deprecation over time is vast and various models have been proposed (Ahn et al., 2020; Marzilli Ericson et al., 2015; Mazur, 1987; Samuelson, 1937; Van den Bos & McClure, 2013). Each of them has its own reasoning and theoretic background. Three different models were included in our analysis. The first one is an *exponential hyperbolic model*. A classic delay discounting theory from the field of economics, initially proposed by Samuelson (1937). It predicts individuals to have time-consistent preferences that are independent of reward amount. The exponential hyperbolic model is defined as

$$\mathrm{V} = A * e^{-1*k*D}. \qquad 2.15$$

V represents the subjective value of an option. A is the amount in Euro of the delayed option, while D is the delay in days. The -1 ensures the exponent to be negative, so the subjective value deprecates. k is a free parameter estimated by the data that determines how quickly the subjective value deprecates. High values for k correspond to fast deprecation. The subjective value is then passed to a sigmoid function to translate the subjective value into a choice for either the smaller immediate reward or the larger delayed reward. An additional parameter beta determines the slope of the sigmoid function, where large beta values correspond to steep slopes and deterministic behaviour, while small betas result in flat sigmoid curves making the translation of a subjective value to a choice ambiguous and increasingly random. Figure 15 illustrates the impact of the parameter beta. It can clearly be seen that a beta value of 1 corresponds to a clear decision of choosing either of the two options, with probabilities close to 1, depending on the subjective evaluation. Only subjective value differences close to 0 have an approximately equal probability of choosing either option. Correspondingly, small beta values represent a more random decision, where regardless of the subjective value differences, the probability for either decision is close to equal.

Figure 15

Relation of Subjective Evaluation and Choice Probability



Note. The displayed sigmoid functions show the probability of choosing option A depending on the value difference between option A and option B. A steep sigmoid function with a large β value indicates deterministic behaviour while a flat sigmoid curve corresponds to random choices independent of value difference.

The second model proposed by Mazur (Mazur, 1987) is also a hyperbolic model containing the same variables; however, it lacks the exponential component. The interesting difference is that contrasting to the first model, this model predicts time-inconsistent preferences. This results in overpredicting subjective value at shorter delays, while underpredicting subjective value at longer delays. Mazur's hyperbolic model is given in equation 2.15.

$$V = \frac{A}{1+k*D} \qquad 2.15$$

The subjective value for Mazur's model is then transformed to a decision for either option by applying a sigmoid function, where the parameter beta determines the slope of the sigmoid the same way it did for the exponential model.

These two models are very similar overall. Both take the amount and the delay into account with two free parameters estimated from the data. On the one hand, the beta determines the slope of the sigmoid function. On the other hand, the deprecation parameter k controls how quickly the subjective value of the larger monetary reward decreases due to the delay.

The third delay discounting model was published recently, claiming to outperform the well-established earlier two. This model developed by Ericson et al. (2015) is named *intertemporal choice heuristic* (ITCH) model. The decisions are made using simple arithmetic comparisons to compare the earlier and later option. This model unlike the earlier two, does not assume an inherent discounting function but compares the two options in absolute and relative terms. The ITCH model is specified as

$$\mathrm{P}(LL) = L(\beta_I + \beta_{xA} * (X_2 - X_1) + \beta_{xR} * \frac{X_2 - X_I}{X^*} + \beta_{tA} * (t_2 - t_1) + \beta_{tR} * \frac{t_2 - t_I}{t^*}) \qquad 2.16$$

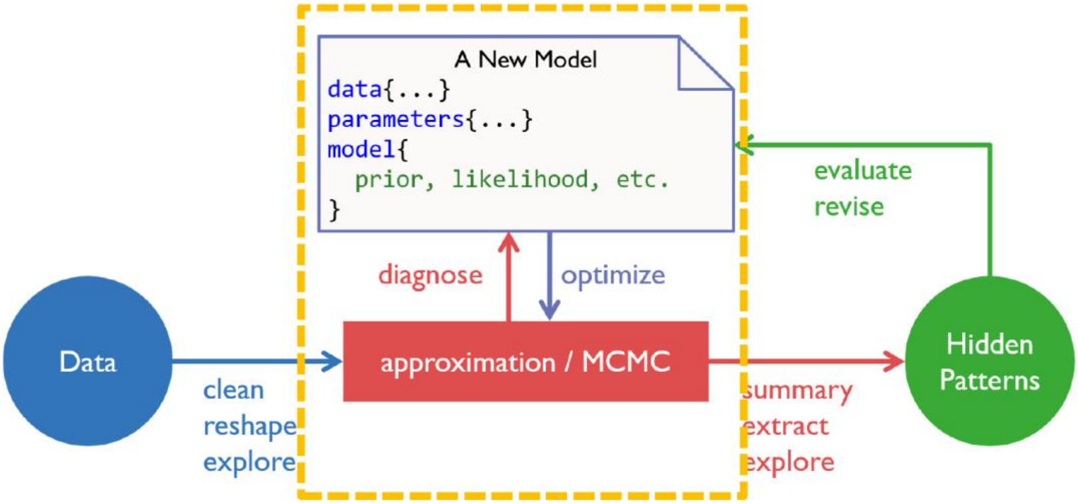
and treats the dimensions of money and time symmetrically. Each option in a binary MEL task takes the form (x, t) where x represents the monetary value of an option and t the time delay. The decision is between a smaller earlier amount (x_I, t_I) and a larger later amount (x_2, t_2). β_I is the intercept term, R stands for “relative” while A means “absolute”. t^* and x^* are the arithmetic means between the two options. x^* can be computed by averaging the amounts between the two options. The same way t^*can be computed by averaging the delays between the two options. Next to the intercept parameter, each of the 4 terms, namely the absolute and relative terms for amount and time, are scaled by individual parameters β. This results in a total of 5 free parameters to be estimated. Unlike the earlier two models, there is no deprecation factor k; the entire evaluation of an option and decision is based on comparisons between the two options.

3.2 Methods

We prepared the data using R and its integrated development environment RStudio (R Core Team, 2019; RStudio Team, 2020). For model implementation we used Stan (Carpenter et al., 2017). Figure 16 shows a conceptual overview of the process and interaction when evaluating a model using MCMC methods.

Figure 16

Pipeline Bayesian Cognitive Modelling



Note. Figure 16 shows the analysis pipeline in Bayesian Cognitive Modelling from data preparation to the MCMC sampling to the model evaluation. Adapted from Ahn et al. (2017).

All models were fitted as hierarchical models. Hierarchical modelling refers to simultaneously estimating group level and subject level parameters. This hierarchical approach leads to shrinkage effects (Gelman et al., 2013) in individual parameter estimates. To achieve shrinkage effects, the subject level parameters inform the estimation of the group levels, and simultaneously the group level parameter estimates affect the estimation of each subject level parameter. Overall, this hierarchical approach results in more stable parameter estimates.

For the MCMC sampling in Stan the default hyperparameters were chosen: In total, 4 chains were used with 2000 iterations. The first 1000 samples were discarded as warm-up. The *warm-up* or *burn-in* deals with the influence of the first proposed value of θ on the posterior distribution and reduces its impact. Another problem in MCMC is its *autocorrelation* (Box et al., 2015, p.54ff): Autocorrelation describes the correlation of a signal with a delayed copy of itself. In a autocorrelated signal there is a similarity between observations as a function of the time lag in-between observations. If autocorrelation is a problem in the drawn samples, *thinning* can improve the posterior. Thinning refers to increasing the MCMC sample size and drawing at regular intervals. For instance, a thinning interval of 3 would correspond to always picking the 3rd accepted value and adding it to the posterior rather than adding every accepted value. However, here autocorrelation did not pose a problem. Thus, thinning was set to 1 and all samples fulfilling the acceptance criterion were included. The complete specification of the models is given in the appendix where the main R script can be found as well as the individual model files.

3.3 Results

All 3 models have been evaluated against the same dataset. To determine the winning model, we used the Widely Applicable Information Criterion (WAIC), an extension of the earlier introduced AIC. WAIC provides estimates of the out of sample accuracy. For this purpose, new participants are sampled from the hierarchical group and new data is generated from these participants. Then the evaluation is based on how well the model can account for the new dataset (Ahn et al., 2017). WAIC is on the information criterion scale; thus, lower values of WAIC indicate better out-of-sample prediction accuracy of the candidate model (Vehtari et al., 2017). Table 1 summarises the 3 included models and their corresponding WAIC scores.

Table 1

Summary of the Model Evaluation

Model	WAIC (SE)	Model weight	Number free parameters	Effective parameters (SE)
Samuelson	1899,2 (81,6)	0,0	2	19,6 (1,9)
Mazur	1844,2 (77,6)	0,898	2	19,0 (2,1)
Ericson	1882,6 (84,0)	0,102	5	24,4 (1,8)

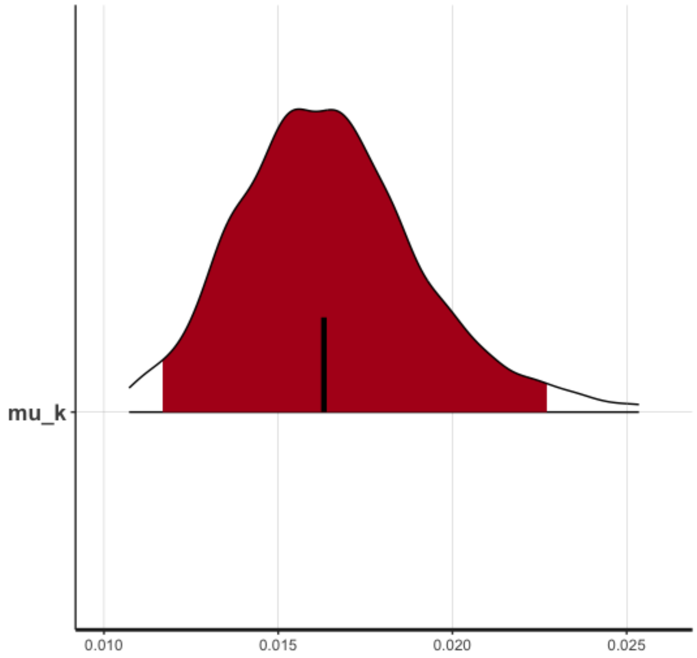
Note. Table 1 shows the Summary of the model fitting. *WAIC* = Widely applicable information criterion. *SE* = Standard error.

The column model weight – similarly to the WAIC – summarizes the relative model support for each model. The model weight is based on the WAIC and the differences among the models in WAIC with respect to the best WAIC sore in the set. For a more detailed introduction of WAIC and model weights see McElreath (2018).

As can be seen in Table 1, from the three proposed models, the model by Mazur (Mazur, 1987) is the winning model with the lowest WAIC score and the highest model weight. This is in line with other results reported in the literature (Van den Bos & McClure, 2013). Purely considering the WAIC score and the standard errors, all models perform similarly well. However, the model weight scores are a clear indication that the hyperbolic model is the winning model. This result goes in line with other model comparisons for MEL tasks (McKerchar et al., 2009). Nevertheless, the dataset used here is a simulated one for illustration purposes. Thus, these results should not be taken as an indication for one model being more or less appropriate than another in explaining these intertemporal decisions, but rather illustrate the Bayesian process of model evaluation and parameter recovery. The latter is the next topic we turn to. After identifying the model which best accounts for the data, the estimated free parameters given by the posterior distribution can be further analysed. These recovered parameters are often of interest to research questions in cognitive science and offer direct interpretability. Here, the research questions determine the analysis and whether the focus is on individual level parameters or the group level. We will focus on parameter estimates across the entire group for the two free parameters in our winning model. Figure 17 shows the posterior density function obtained by MCMC sampling for the deprecation parameter k at the group level.

Figure 17

Posterior Distribution for the Deprecation Parameter k



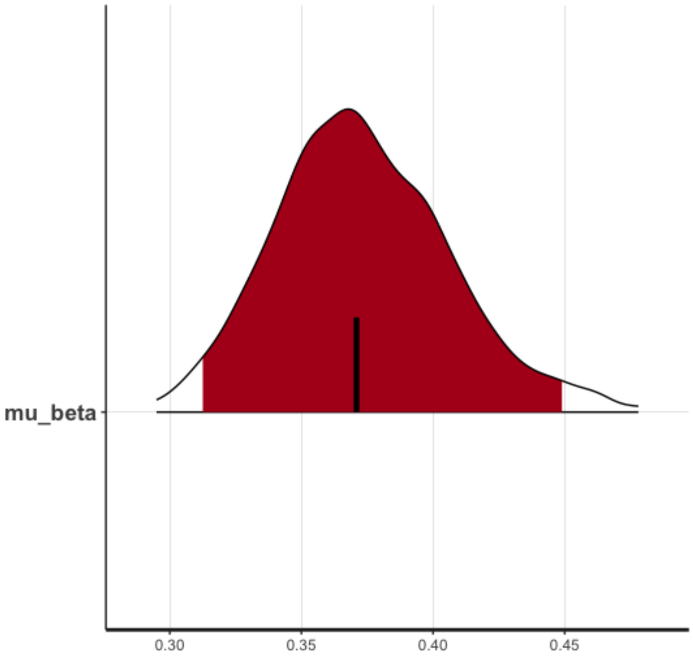
Note. Figure 17 shows the posterior distribution for the deprecation parameter k from the winning hyperbolic model. The vertical black bar indicates the distributions' mode. The red area indicates the 95% credible interval, while the horizontal black line shows the 99% credible interval.

Based on the posterior density function several conclusions can be drawn. The "best guess" we can give for our true parameter value corresponds to the mode of the distribution, which in the present example is at about 0.017. However, next to the best guess the density function also represents "how certain" we can be about our parameter estimates. The wider the distribution the more uncertain the estimates of the parameter value. In Figure 17, next to the distributions' mode, two more commonly reported metrics are displayed. The red coloured area corresponds to 95% of the mass of the posterior density function. It can be interpreted as a 95% percent probability of the true value being in this range, which is also called the *credible interval* (Lee & Wagenmakers, 2014). Next to the 95% credible interval the 99% credible interval is displayed as the horizontal black line beneath the curve. This line displays the tails of the distribution and gives a certainty estimate for the parameter estimation. Furthermore, the posterior density function enables the calculation of the probability that the true parameter value is in a certain parameter value range by estimation of the surface area under the function in the parameter value range of interest. Here, the value of 0.017 indicates a slow deprecation of the subjective value for the present example on the MEL task.

Next to the deprecation parameter, the slope of the logistic regression function was estimated and is represented by the parameter β. Figure 18 shows the posterior probability density function for the parameter β again at the group. Here, the mode is the best guess for our parameter value which corresponds to β = 0.37. The red area corresponds to 95% of the mass of the probability density function, showing the certainty of our parameter estimate. The black line running horizontally corresponds to 99% of the mass, showing the tails of the function.

Figure 18

Posterior Distribution for the Slope Parameter β



Note. Figure 18 shows the posterior distribution for the slope parameter β from the winning hyperbolic model. The vertical black bar indicates the distributions' mode. The red area indicates the 95% credible interval, while the horizontal black line shows the 99% credible interval.

4 Discussion

In the last chapter of this thesis, the case study will be connected to the earlier introduced concepts of computational modelling in cognitive science. Afterwards limitations and assumptions in (Bayesian) cognitive modelling will be addressed to then conclude with what I believe computational cognitive modelling and Bayesian methods have to offer to cognitive science.

4.1 Situating the Delay Discounting Example

The models introduced in the previous chapter describe and explain the experimental data on delay discounting. All three models instantiate theories of the underlying mechanisms in delay discounting and map the effect of temporal delay on to the subjective deprecation of value. Some introduce additional variables or different weightings of individual predictors, nevertheless all provide a statistical description of delay discounting that can be applied to explain the collected data. In addition, next to the mere statistical description they further explicate the relevant variables essential for the subjective value deprecation as a result of time delay. This is different to other purely statistical approaches finding patterns in data while not explicating the individual contributions of predictors or being fully opaque. Overall, the purpose of models such as the in chapter 3 introduced depend on the assumptions of the modeling scientist. They can be taken to formalize a mathematical description of the observed data while explicating relevant variables or can be theories about the ongoing cognitive processes that bring about the subjective deprecation of value. The earlier being a mathematical description of behavior the latter being a *cognitive process model*. One example for the latter is the in the introduction described model of categorization. Here the proposal is that participants go about categorizing by actively calculating a dissimilarity metric.

To estimate the model parameters MCMC sampling was applied as a method. Other methods like root mean squared error minimization or maximum likelihood estimation would have been possible alternatives to estimate the parameters. One reason for choosing MCMC sampling as a method is the generated posterior distribution which allows for an interpretation of the (un)certainty around the estimated parameters. Furthermore, there is the open question whether MCMC sampling might be a model for how humans go about solving problems employing a MCMC-like sampling process as for example in perceptual multistability (Gershman et al., 2009).

Multiple complementing approaches can be applied to evaluate models. On the one hand a model can be analyzed theoretically, whether it makes reasonable predictions i.e., there cannot be a negative memory capacity as introduced in chapter 2.5. Furthermore, simulations allow to test a model’s behavior. These can be complemented with empirical data, ideally across tasks or populations to assess their generalizability. All models introduced were only evaluated against one specific task. Thus, the conclusions cannot take into account how well the models generalize or potentially fail to do so. Furthermore, the models were only evaluated quantitatively comparing their fit to the data while controlling for complexity, given by model weight and WAIC (Vehtari et al., 2017). Based on the here presented analyses, the conclusion would be in favour of the exponential model by (Mazur, 1987). However, this conclusion comes with the caveat of omitting theoretical conceptual limitations or insights generated by other experiments in delay discounting research. Since the case study here served as a vehicle for illustrating and understanding Bayesian computational modelling, the ongoing discussion around delay discounting will not be explored further. Instead, computational modelling as a tool in cognitive science will be discussed.

One introduced concept is the levels of explanation at which the cognitive phenomenon is addressed as introduced in chapter 1.4. Models can either start out at the hardware implementational level taking a bottom-up approach or at the higher cognitive level, working their way down in a top-down fashion. The in the case study introduced models take a top-down approach starting out at the higher cognitive level specifying the functions used to solve the problem of delay discounting. By specifying representations of the relevant variables and operations over them for solving the delay discounting problem, the models can be understood as classical cognitive models. Bayesian cognitive models are not ordinarily construed to be a classical cognitive model (Samuels, 2019). However, Samuels (2019) describes them to be classical good old fashioned computational models (CCTM) with some twist. CCTM portrays the mind literally as a “classical computational system an interpretable, formal symbol manipulator – of some sort” (Samuels, 2019). The difference of the Bayesian approach being that it does not have a doctrinal commitment to the classical computational theory of mind (Samuels, 2019). However, the specification of representations of relevant variables and the operations upon those variables share the approach with classical cognitive models. Nevertheless, the tools applied in Bayesian cognitive modelling like MCMC simulations to approximate parameters and the ideal solving of problems differentiates the Bayesian approach from other classical approaches.

4.2 Criticisms to (Bayesian) cognitive models

Bayesian cognitive modelling shares core assumptions with classical cognitive models and thus faces similar criticisms. Additionally, there are special limitations to Bayesian approaches. Both, the general limitations to classical cognitive models and the criticism Bayesian methods face will be addressed next.

Generally, MCMC simulations applied to obtain model parameters and evaluate the constructed models against data are relatively uncontroversial as a tool. They are in line with other techniques like least squares estimation and Maximum Likelihood estimations. Here, they are additionally offering the advantage of the posterior distribution containing information about the *certainty* of the estimation. Criticisms of MCMC sampling mostly rest on pragmatic limitations of the technique with the two biggest limitations being the computational requirements to obtain parameter estimates and additionally the conceptual complexity of the methods which make their adoption challenging (Levy, 2009; Robert et al., 2018).

The assumptions taken and the models constructed in classical cognitive modelling like the ones introduced in the previous chapter are much more controversial (Purves, 2021). Models built at the higher levels in Marr's hierarchy start out with the cognitive functions. They focus on the problem and how it is solved in functional terms while not directly addressing how the proposed mechanisms solving the problem are implemented at the hardware level. This is a common criticism classical cognitive models face, as they often do not address the implementational level or cannot fully map their proposed representations and computations onto mechanisms found in the brain. This criticism is faced by both classical computational cognitive models as well as connectionist models, but it is more prominent in the former as the connectionist approach is inspired by its biological counterpart and thus conceived to be more biologically plausible, although backpropagation based approaches are criticized to be incomplete and cannot account for all human learning abilities (Farrell & Lewandowsky, 2018, p.364; Kriegeskorte & Douglas, 2018). Thus, both top-down and bottom-up approaches face the challenge of biological realization and are at best loosely coupled to their target system, the brain.

The distinction between bottom-up and top-down approaches can be thought of as a competition for explanatory primacy. However, the different approaches from different fields do not have to be thought of as competing against one another. Kriegeskorte & Douglas (2018) link the varying disciplines to Marr's tri level hierarchy: Cognitive Science working on the computational level decomposing cognition into components, taking a top-down approach.

Artificial Intelligence working on the algorithmic level, building representations and algorithms, and Computational Neuroscience composing neural building blocks into representations and algorithms. However, rather than being in competition they view them in a complementary role and try to reconcile them by promoting the field of computational cognitive neuroscience. This new field is supposed to bridge and integrate approaches from different fields whereas the functional top-down approach together with the implementational bottom-up approach mutually constrain one another. The overall aim is to construct task-performing computational models that explain how cognition arises from neurobiologically plausible dynamic components. This new branch of interdisciplinary research is a growing field with enthusiasm to advance cognition research by interdisciplinary collaboration as voiced in a recent opinion paper titled: Cognitive Neuroscience: A new conference for an emerging discipline (Naselaris et al., 2018).

Such an approach is also proposed as a way forward for the field of delay discounting research as suggested by Van den Bos & McClure (2013). While acknowledging hyperbolic discounting models as eminently useful quantitative measures, they criticize it for it’s lacking ability as a descriptive psychological model of the cognitive processes that produce intertemporal preferences and emphasize the importance of a general model of time discounting. As one way forward they propose to incorporate insights from cognitive neuroscience. They propose that a neuroscience-based theory might be able to reconcile the empirical data on time preferences and disparate theoretical models.

Constructing computational cognitive models grounded in neuroscientific insights is one way to address criticisms of biological plausibility. However, computational cognitive models face other, more general objections: These objections include the a priori objection put forward by John Searle (Cole, 2020). His criticism expressed in the Chinese room argument can be summarized as computational models of the mind being inappropriate because performing the right computations itself is insufficient for cognitive capacities such as understanding (Cole, 2020). His thought experiment imagines a non-Chinese speaking person in a room into which Chinese messages are passed. According to a set of rules the person responds to the incoming messages in Chinese. The person does not understand Chinese but with the help of the rule set can respond appropriately to the incoming message in Chinese. Now, the question is whether the person understands Chinese or rather just simulates the ability to understand Chinese. Searle’s answer is the latter. For him the ability to produce syntactically correct sentences is insufficient for a semantic understanding. From there he generalizes his argument based on language and language understanding to cognition as a whole. Concluding that “running a program no matter how good is insufficient for cognition” (Samuels, 2019, p.114). The debate the Chinese room sparked is long and would make for more than a thesis by itself. One reply is that classicists do not claim that executing the right program by itself is sufficient for thought. The claim is that cognitive processes are computational processes operating on semantically evaluable representations while leaving the option open that semantic properties are determined by something other than the computational role, like causal relations to the environment (Fodor, 1990; Samuels, 2019). Following this argument, Searle’s conclusion would be compatible with the theories put forward describing cognition in computational terms. Here, we will move on to the next criticism of computational models of cognition. A more thorough summary of the discussion around the Chinese room argument and its’ responses can be found in Cole (2020).

Another family of criticisms maintains that the classical paradigm is insufficient to explain various psychological phenomena. This can either refer to cognitive phenomena that yet have not been sufficiently addressed or seemingly cannot be addressed in principle using computational models, e.g., creativity or phenomenally conscious states such as perceptual experiences or emotions. This is in line with Nagel’s statement that there is “something that it is like to be” in those states (Nagel, 1974, p.436). Here, the claim is that classical computational models cannot provide a satisfactory account of how organisms can have these phenomenal states, which cannot be plausibly characterized in terms of functional or computational roles (Chalmers, 2004). Nevertheless, despite the lacking account of phenomenally conscious states it does not render computational models useless as will be addressed further in the offerings of computational cognitive modelling in section 4.3.

One more criticism is the increase in abstraction. In the process of modelling, we necessarily increase the abstraction. The ideal model is simple with a large explanatory power but as a model is simpler the abstraction increases and necessarily more details are omitted. Closely tied to the problem of the trade-of between abstraction and explanatory power is the problem of validating abstract models. Assuming there are two (or more) developed models performing similarly well as indicated by the quantitative measures, how can we know whether the proposed model actually captures the way the problem is solved rather than an analogous process with the same outcome? The winning computational model as well as the collected behavioural data potentially align. However, there is no guarantee that the by the participant implemented strategy is the same as the proposed solution by the model. This phenomenon is known as the *identification problem, which* describes different sets of parameters that result in the same outcome. A simple example would be the adding of two numbers like 10 and 19. The solution 29 can be obtained by adding 10 and adding another 9 or by adding 20 and subtracting 1. These two and potential other ways come to the same conclusion via different routes.

One way to cope with the problem of validating abstract models is via carefully designed experiments that rule out alternative explanations. Here the behavioural data is used to challenge the proposed model and rule out alternative explanations. This interplay and the clear falsifiability of models if they cannot account for certain behavioural findings is in fact one of the biggest strengths in computational cognitive modelling compared to alternative models lacking a clear-cut falsification criterion. Next to evaluating the models against behavioural experiments it becomes more and more common to simultaneously develop a computational cognitive model with an analysis of the along going neural activity. On the one hand this enables to explore how the proposed models might be implemented in the brain as well as to rule out others and, on the other hand, to evaluate the model against the knowledge of neural activity, to see whether the predictions of the model and the activated brain areas align. However, this line of research is still very young and the methodological access to the brain rarely exceeds a functional activation of a given brain area (Farrell & Lewandowsky, 2018). Nevertheless, there are first studies pursuing this approach like for example Zhang & Gläscher (2020). Furthermore, there is an attempt to integrate the different disciplines, working towards a joint framework (Laird et al., 2017). At this intersection of computational cognitive modelling and neuroscience much progress has been made over the recent years and it currently is a field with a lot of attention (Kriegeskorte & Douglas, 2018; Naselaris et al., 2018).

4.3 Contributions of Computational cognitive modelling to Cognitive Science

The last part of the thesis will deal with the offerings computational cognitive modelling brings to cognitive science. The first contribution of computational models to cognitive science research are *descriptive* models illustrating statistical relationships between variables. The example given in chapter 1.2 was skill acquisition over trials (Figure 1, p.5). While not making assumptions about the underlying processes, these models still offer a valuable contribution and enable a researcher to better understand the statistical relations. This "first level" explanation, as it was termed by Lewandowsky & Oberauer (2018), goes from individual observations to statistical generalizations and was pushed forward with the recent increase of attention on machine learning models. Machine learning models have been applied to various research areas in cognitive science to establish statistical generalizations like for example in clinical research (Dwyer et al., 2018), differential psychology (Bleidorn et al., 2017) or neuroscience (Huys et al., 2016; Kriegeskorte & Douglas, 2018). Across phenomena machine learning gained popularity and sparked excitement to the extent that even the stance of theory is put into question by some, promoting a purely data driven approach fueled by advances in machine learning, large computing resources, and increasing data (Mazzocchi, 2015).

When moving beyond a statistical description and constructing cognitive *process models*, the in chapter 1.2 – 1.4 discussed assumptions and limitations have to be addressed. However, although classical computational models of the mind might be deficient they are still helpful and explanatory at some level of granularity (Samuels, 2019). Samuel's (2019) comparison is electrical circuit theory being explanatory in cellular neuroscience even though almost no one maintains that neurons are just electrical circuits. In that way the produced models are vehicles in order to think about the phenomenon of interest.

Cognitive modelling acts as a *cognitive prosthesis* by avoiding under-specification (Lewandowsky & Oberauer, 2018). It requires the researchers to be explicit about the assumptions they make (Samuels, 2019). Next to that, they generate precise predictions which can be tested against behavioural data and are subject to falsification and thus, well suited for the scientific method (Samuels, 2019). Therefore, computational modelling is one tool to tackle the replicability crisis and can enable more coherent theory building (Guest & Martin, 2021).

Cognitive models can be evaluated against behavioural data and afford a bridging between behavioural observational data and biological data (Lewandowsky & Oberauer, 2018). All computational cognitive models face the criticism of biological plausibility. One the one hand, connectionist models may be loosely inspired by their biological counterpart, nevertheless their similarity to a biological neural network and it's cognitive functionalities is imperfect (Pulvermüller et al., 2021). Similarly classical cognitive models face the criticism about how the proposed functions are realized in the brain (Love, 2015, 2021). One way forward as explained in chapter 1.4 might be the field of computational cognitive science reconciling a bottom-up and a top-down approach.

The given arguments generally apply to computational cognitive modelling. In the thesis the focus was on Bayesian methods in cognitive science. There are many models built assuming that the brain performs at least some tasks in (Bayesian-) like ways which here were called Bayesian models of the mind. This metaphor has proven to be useful and productive, resulting in many models, e.g., for perceptual tasks (Hohwy et al., 2008) or decision-making under uncertainty (Lloyd & Leslie, 2013). It is a useful metaphor for tasks that require the representation of uncertainty or when it comes to integrating new with already existing information. Whether the brain performs these tasks in a Bayesian way remains to be seen as the biological implementation is still not understood but I personally am convinced that the approach will continue to be productive. However, eventually the cognitive models inspired by Bayes' theorem will have to be tied to theories and empirical data of how these cognitive functions are implemented in the brain. For this question there are first theories but no consensus yet (Friston, 2010; Gallagher et al., 2021; Knill & Pouget, 2004).

The second Bayesian method introduced in this thesis are Bayesian cognitive models. Here, there is no assumption of the brain working in Bayes' like ways. It is rather about the tool of MCMC-based simulations to evaluate models against collected data and generating posterior distributions of model parameters, which again have a psychological interpretation. The method itself has several advantages like its versatility. There is a large variety of models which can be built and evaluated against data using MCMC sampling techniques. Furthermore, the Bayesian approach has a very intuitive way of representing *uncertainty* about the generated parameter estimates and the evaluated models. This again allows the modelling researcher to adjust the conclusions taken away from the models.

5 Conclusion

In this thesis we explore computational modelling as a tool in cognitive science to study cognition. Computational models have been described as a cognitive prosthesis to the cognitive scientist and serve various purposes (Farrell & Lewandowsky, 2018). The differences between *descriptive* and *explanatory* computational models are addressed as well as the different levels of analysis at which computational cognitive models are constructed. In the recent past the Bayes' theorem is seeing an increased usage in cognitive modelling. Three different applications were introduced: Bayesian statistical analysis, Bayesian models of the mind and Bayesian cognitive models. The latter was examined in detail with an example from delay discounting research. Bayesian cognitive modelling in the form of MCMC sampling allows to compare models against each other and estimate model parameters in the light of collected data. Additionally, they are explicit in their interpretation by generating probability distributions explicating the (un-)certainty about the estimated parameters. Two frequent criticisms are the computational requirements to perform MCMC sampling as well as the conceptual complexity making it time-consuming for scientists to adopt the method. Besides such practical considerations, Bayesian cognitive modelling shares the limitations of other computational models like triviality arguments or lacking semantic understanding as voiced in the form of the Chinese room argument. Thus, the appropriateness of Bayesian cognitive modelling as tool depends on the task at hand as well as the theoretical cognitive framework the modelling is related to. Irrespective of the philosophical discussion around the computational theory of mind, Bayesian cognitive models provide one decisive advantage, which makes them a valuable tool for cognitive science. They demand the cognitive scientist to conceptually analyze and explicate theories in the form of quantitative models, enabling quantitative evaluation and communication among a community of scientists.

6 References

Ahn, W.-Y., Gu, H., Shen, Y., Haines, N., Hahn, H. A., Teater, J. E., Myung, J. I., & Pitt, M. A. (2020). Rapid, precise, and reliable measurement of delay discounting using a Bayesian learning algorithm. *Scientific Reports*, *10*(1), 12091. https://doi.org/10.1038/s41598-020-68587-x

Ahn, W.-Y., Haines, N., & Zhang, L. (2017). Revealing Neurocomputational Mechanisms of Reinforcement Learning and Decision-Making With the hBayesDM Package. *Computational Psychiatry*, *1*, 24–57. https://doi.org/10.1162/CPSY_a_00002

Akaike, H. (1998). Information theory and an extension of the maximum likelihood principle. In *Selected papers of Hirotugu Akaike* (pp. 199–213). Springer.

Anderson, J. R. (2013). *The adaptive character of thought*. Psychology Press.

Bayes, T., & Price (1763). An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, F. R. S. communicated by Mr. Price, in a letter to John Canton, A. M. F. R. S. *Philosophical Transactions of the Royal Society of London*, *53,* 370–418. https://doi.org/10.1098/rstl.1763.0053

Bechtel, W. (1994). Levels of description and explanation in cognitive science. *Minds and Machines*, *4*(1), 1–25. https://doi.org/10.1007/BF00974201

Bechtel, W., & Abrahamsen, A. (1991). *Connectionism and the mind: An introduction to parallel processing in networks.* Basil Blackwell.

Bickle, J. (2020). Multiple Realizability. In E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy* (Summer 2020). Metaphysics Research Lab, Stanford University. https://plato.stanford.edu/archives/sum2020/entries/multiple-realizability/
(Date of Access: 22.1.2022)

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Bleidorn, W., Hopwood, C. J., & Wright, A. G. (2017). Using big data to advance personality theory. *Current Opinion in Behavioral Sciences*, *18,* 79–82. https://doi.org/10.1016/j.cobeha.2017.08.004

Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: Forecasting and control*. John Wiley & Sons.

Bozdogan, H. (2000). Akaike’s information criterion and recent developments in information complexity. *Journal of Mathematical Psychology*, *44*(1), 62–91.

Bröder, A., & Schütz, J. (2009). Recognition ROCs are curvilinear—Or are they? On premature arguments against the two-high-threshold model of recognition. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *35*(3), 587.

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., & Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of Statistical Software*, *76*(1).

Chalmers, D. J. (2004). How Can We Construct a Science of Consciousness? In M. S. Gazzaniga (Ed.), *The Cognitive Neurosciences.* (pp. 1111–1119). MIT Press.

Chater, N., Tenenbaum, J. B., & Yuille, A. (2006). *Probabilistic models of cognition: Conceptual foundations*. Elsevier.

Chechile, R. A. (1998). Reexamining the goodness-of-fit problem for interval-scale scores. *Behavior Research Methods, Instruments, & Computers*, *30*(2), 227–231. https://doi.org/10.3758/BF03200648

Chechile, R. A. (1999). A vector-based goodness-of-fit metric for interval-scaled data. *Communications in Statistics – Theory and Methods*, *28*(2), 277–296. https://doi.org/10.1080/03610929908832298

Clark, A. (2013). Whatever next? Predictive brains, situated agents, and the future of cognitive science. *Behavioral and Brain Sciences*, *36*(3), 181–204. https://doi.org/10.1017/S0140525X12000477

Clark, A. (2015). *Surfing uncertainty: Prediction, action, and the embodied mind*. Oxford University Press.

Cole, D. (2020). The Chinese Room Argument. In E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy* (Spring 2020). Metaphysics Research Lab, Stanford University. https://plato.stanford.edu/archives/spr2020/entries/chinese-room/
(Date of Access: 22.1.2022)

Colombo, M., & Knauff, M. (2020). Editors' Review and Introduction: Levels of Explanation in Cognitive Science: From Molecules to Culture. *Topics in Cognitive Science*, *12*(4), 1224–1240. https://doi.org/10.1111/tops.12503

Cooper, R. P. (2007). The role of falsification in the development of cognitive architectures: Insights from a Lakatosian analysis. *Cognitive Science*, *31*(3), 509–533.

Cramer, A. O. J., Ravenzwaaij, D. van, Matzke, D., Steingroever, H., Wetzels, R., Grasman, R. P. P. P., Waldorp, L. J., & Wagenmakers, E.-J. (2016). Hidden multiplicity in exploratory multiway ANOVA: Prevalence and remedies. *Psychonomic Bulletin & Review*, *23*(2), 640–647. https://doi.org/10.3758/s13423-015-0913-5

Culbertson, J., & Smolensky, P. (2012). A Bayesian Model of Biases in Artificial Language Learning: The Case of a Word-Order Universal. *Cognitive Science*, *36*(8), 1468–1498. https://doi.org/10.1111/j.1551-6709.2012.01264.x

Dwyer, D. B., Falkai, P., & Koutsouleris, N. (2018). Machine Learning Approaches for Clinical Psychology and Psychiatry. *Annual Review of Clinical Psychology*, *14*(1), 91–118. https://doi.org/10.1146/annurev-clinpsy-032816-045037

Eliasmith, C. (2013). *How to build a brain: A neural architecture for biological cognition*. Oxford University Press.

Eliasmith, C., Stewart, T. C., Choo, X., Bekolay, T., DeWolf, T., Tang, Y., & Rasmussen, D. (2012). A Large-Scale Model of the Functioning Brain. *Science, 338*(6111), 1202–1205. https://doi.org/10.1126/science.1225266

Farrell, S. (2012). Temporal clustering and sequencing in short-term memory and episodic memory. *Psychological Review*, *119*(2), 223.

Farrell, S., & Lewandowsky, S. (2018). *Computational Modeling of Cognition and Behavior*. Cambridge University Press. https://doi.org/10.1017/9781107109995

Fidler, F., & Wilcox, J. (2021). Reproducibility of Scientific Results. In E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy* (Summer 2021). Metaphysics Research Lab, Stanford University. https://plato.stanford.edu/archives/sum2021/entries/scientific-reproducibility/
(Date of Access: 22.1.2022)

Fodor, J. A. (1990). *A theory of content and other essays.* MIT Press.

Frederick, S., Loewenstein, G., & O'donoghue, T. (2002). Time discounting and time preference: A critical review. *Journal of Economic Literature*, *40*(2), 351–401.

Frigg, R., & Hartmann, S. (2020). Models in Science. In E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy* (Spring 2020). Metaphysics Research Lab, Stanford University. https://plato.stanford.edu/archives/spr2020/entries/models-science/
(Date of Access: 22.1.2022)

Friston, K. (2010). The free-energy principle: A unified brain theory? *Nature Reviews Neuroscience*, *11*(2), 127–138.

Fum, D., Missier, F. D., & Stocco, A. (2007). The cognitive modeling of human behavior: Why a model is (sometimes) better than 10,000 words. *Cognitive Systems Research*, *8*(3), 135–142. https://doi.org/10.1016/j.cogsys.2007.07.001

Gallagher, S., Hutto, D., & Hipólito, I. (2021). Predictive Processing and Some Disillusions about Illusions. *Review of Philosophy and Psychology*. https://doi.org/10.1007/s13164-021-00588-9

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

Gershman, S. J., Vul, E., & Tenenbaum, J. B. (2009). Perceptual multistability as Markov Chain Monte Carlo inference. *Neural Information Processing Systems (NIPS)*. https://dspace.mit.edu/handle/1721.1/112788

Gopnik, A., & Tenenbaum, J. B. (2007). Bayesian networks, Bayesian learning and cognitive development. *Developmental Science*, *10*(3), 281–287. https://doi.org/10.1111/j.1467-7687.2007.00584.x

Griffiths, T. L., Chater, N., Kemp, C., Perfors, A., & Tenenbaum, J. B. (2010). Probabilistic models of cognition: Exploring representations and inductive biases. *Trends in Cognitive Sciences*, *14*(8), 357–364. https://doi.org/10.1016/j.tics.2010.05.004

Guest, O., & Martin, A. E. (2021). How Computational Modeling Can Force Theory Building in Psychological Science. *Perspectives on Psychological Science*, 1745691620970585. https://doi.org/10.1177/1745691620970585

Hauser, T. U., Fiore, V. G., Moutoussis, M., & Dolan, R. J. (2016). Computational psychiatry of ADHD: neural gain impairments across Marrian levels of analysis. *Trends in Neurosciences*, *39*(2), 63–73.

Heathcote, A., Brown, S., & Mewhort, D. J. K. (2000). The power law repealed: The case for an exponential law of practice. *Psychonomic Bulletin & Review*, *7*(2), 185–207. https://doi.org/10.3758/BF03212979

Held, L., & Bové, D. S. (2014). *Applied Statistical Inference: Likelihood and Bayes*. Springer. https://doi.org/10.1007/978-3-642-37887-4

Hohwy, J. (2013). *The predictive mind*. Oxford University Press.

Hohwy, J., Roepstorff, A., & Friston, K. (2008). Predictive coding explains binocular rivalry: An epistemological review. *Cognition*, *108*(3), 687–701. https://doi.org/10.1016/j.cognition.2008.05.010

Horga, G., Schatz, K. C., Abi-Dargham, A., & Peterson, B. S. (2014). Deficits in Predictive Coding Underlie Hallucinations in Schizophrenia. *Journal of Neuroscience*, *34*(24), 8072–8082. https://doi.org/10.1523/JNEUROSCI.0200-14.2014

Huys, Q. J. M., Maia, T. V., & Frank, M. J. (2016). Computational psychiatry as a bridge from neuroscience to clinical applications. *Nature Neuroscience*, *19*(3), 404–413. https://doi.org/10.1038/nn.4238

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112). Springer.

Jaynes, E. T. (2003). *Probability theory: The logic of science*. Cambridge University Press.

Jones, M., & Love, B. C. (2011). Bayesian Fundamentalism or Enlightenment? On the explanatory status and theoretical contributions of Bayesian models of cognition. *Behavioral and Brain Sciences*, *34*(4), 169–188. https://doi.org/10.1017/S0140525X10003134

Kelso, J. S. (1995). *Dynamic patterns: The self-organization of brain and behavior*. MIT Press.

Kersten, D., & Yuille, A. (2003). Bayesian models of object perception. *Current Opinion in Neurobiology, 13*(2), 150–158.

Knill, D. C., & Pouget, A. (2004). The Bayesian brain: The role of uncertainty in neural coding and computation. *Trends in Neurosciences*, *27*(12), 712–719. https://doi.org/10.1016/j.tins.2004.10.007

Kotseruba, I., & Tsotsos, J. K. (2020). 40 years of cognitive architectures: Core cognitive abilities and practical applications. *Artificial Intelligence Review*, *53*(1), 17–94. https://doi.org/10.1007/s10462-018-9646-y

Kriegeskorte, N., & Douglas, P. K. (2018). Cognitive computational neuroscience. *Nature Neuroscience*, *21*(9), 1148–1160. https://doi.org/10.1038/s41593-018-0210-5

Kruschke, J. (2014). *Doing Bayesian data analysis: A tutorial with R, JAGS, and Stan*. Academic Press.

Kruschke, J. K. (2010). What to believe: Bayesian methods for data analysis. *Trends in Cognitive Sciences*, *14*(7), 293–300.

Laird, J. E. (2012). *The Soar cognitive architecture*. MIT Press.

Laird, J. E., Lebiere, C., & Rosenbloom, P. S. (2017). A Standard Model of the Mind: Toward a Common Computational Framework across Artificial Intelligence, Cognitive Science, Neuroscience, and Robotics. *AI Magazine*, *38*(4), 13–26. https://doi.org/10.1609/aimag.v38i4.2744

Lambert, B. (2018). *A student's guide to Bayesian statistics*. Sage.

Lee, M. D. (2018). Bayesian Methods In Cognitive Modeling. In E.-J. Wagenmakers & J. T. Wixted (Eds.), *Stevens' Handbook of Experimental Psychology and Cognitive Neuroscience* (pp. 37–84). Blackwell Publishing.

Lee, M. D., & Wagenmakers, E.-J. (2014). *Bayesian cognitive modeling: A practical course*. Cambridge University Press.

Lehman, J. F., Laird, J. E., & Rosenbloom, P.(1996). A gentle introduction to Soar, an architecture for human cognition. *Invitation to Cognitive Science*, *4*, 212–249.

Levin, J. (2018). Functionalism. In E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy* (Fall 2018). Metaphysics Research Lab, Stanford University.
https://plato.stanford.edu/archives/fall2018/entries/functionalism/
(Date of Access: 22.1.2022)

Levy, R. (2009, December 30). *The Rise of Markov Chain Monte Carlo Estimation for Psychometric Modeling* [Review Article]. Journal of Probability and Statistics; Hindawi.
https://doi.org/10.1155/2009/537139

Lewandowsky, S., & Farrell, S. (2011). *Computational modeling in cognition: Principles and practice*. Sage Publications.

Lewandowsky, S., & Oberauer, K. (2018). Computational Modeling In Cognition And Cognitive Neuroscience. In E.-J. Wagenmakers & J. T. Wixted (Eds.), *Stevens' Handbook of Experimental Psychology and Cognitive Neuroscience* (pp. 1–36). Blackwell Publishing.

Lindley, D. V. (1972). *Bayesian statistics, a review* (Vol. 2). SIAM.

Lloyd, K., & Leslie, D. S. (2013). Context-dependent decision-making: A simple Bayesian model. *Journal of the Royal Society Interface*, *10*(82). https://doi.org/10.1098/rsif.2013.0069

Love, B. C. (2015). The algorithmic level is the bridge between computation and brain. *Topics in Cognitive Science*, *7*(2), 230–242.

Love, B. C. (2021). Levels of biological plausibility. *Philosophical Transactions of the Royal Society B*, *376*(1815), 20190632.

Marcus, G. F. (2001). *The algebraic mind: Integrating connectionism and cognitive science*. Cambridge, Mass. MIT Press.

Marr, D. (1982). Vision. A computational investigation into the human representation and processing of visual information. In *Vision. A computational investigation into the human representation and processing of visual information*. W.H. Freeman and Company (1982) & MIT Press (2010).

Marzilli Ericson, K. M., White, J. M., Laibson, D., & Cohen, J. D. (2015). Money earlier or later? Simple heuristics explain intertemporal choices better than delay discounting does. *Psychological Science*, *26*(6), 826–833.

Mazur, J. E. (1987). An adjusting procedure for studying delayed reinforcement. In *The effect of delay and of intervening events on reinforcement value* (pp. 55–73). Lawrence Erlbaum Associates, Inc.

Mazzocchi, F. (2015). Could Big Data be the end of theory in science? *EMBO Reports*, *16*(10), 1250–1255. https://doi.org/10.15252/embr.201541001

McClelland, J. L., Botvinick, M. M., Noelle, D. C., Plaut, D. C., Rogers, T. T., Seidenberg, M. S., & Smith, L. B. (2010). Letting structure emerge: Connectionist and dynamical systems approaches to cognition. *Trends in Cognitive Sciences*, *14*(8), 348–356. https://doi.org/10.1016/j.tics.2010.06.002

McElreath, R. (2018). *Statistical rethinking: A Bayesian course with examples in R and Stan*. Chapman and Hall/CRC.

McKerchar, T. L., Green, L., Myerson, J., Pickford, T. S., Hill, J. C., & Stout, S. C. (2009). A comparison of four models of delay discounting in humans. *Behavioural Processes*, *81*(2), 256–259.

Mendonça, D., Curado, M., & Gouveia, S. S. (2020). *The philosophy and science of predictive processing*. Bloomsbury Publishing.

Myung, J. I., & Pitt, M. A. (2018). Model Comparison In Psychology. In E.-J. Wagenmakers & J. T. Wixted (Eds.), *Stevens' Handbook of Experimental Psychology and Cognitive Neuroscience* (pp. 85–118). Blackwell Publishing.

Nagel, T. (1974). What is it like to be a bat? *The Philosophical Review*, *83*(4), 435–450.

Narayanan, S., Jurafsky, D., & others. (1998). Bayesian models of human sentence processing. *Proceedings of the Twelfth Annual Meeting of the Cognitive Science Society*, 752–757.

Naselaris, T., Bassett, D. S., Fletcher, A. K., Kording, K., Kriegeskorte, N., Nienborg, H., Poldrack, R. A., Shohamy, D., & Kay, K. (2018). Cognitive Computational Neuroscience: A New Conference for an Emerging Discipline. *Trends in Cognitive Sciences*, *22*(5), 365–367. https://doi.org/10.1016/j.tics.2018.02.008

Newell, A. (1973). You can't play 20 questions with nature and win: Projective comments on the papers of this symposium. In *In W. G. Chase (Ed.), Visual information processing* (pp. 283–308). New York, NY Academic Press.

Newell, A. (1994). *Unified Theories of Cognition*. Harvard University Press.

Niv, Y., & Langdon, A. (2016). Reinforcement learning with Marr. *Current Opinion in Behavioral Sciences*, *11*, 67–73. https://doi.org/10.1016/j.cobeha.2016.04.005

Nosofsky, R. M. (1986). Attention, similarity, and the identification–categorization relationship. *Journal of Experimental Psychology: General*, *115*(1), 39.

Nosofsky, R. M. (2011). The generalized context model: An exemplar model of classification. In *Formal approaches in categorization* (pp. 18–39). Cambridge University Press. https://doi.org/10.1017/CBO9780511921322.002

Oberauer, K., & Lewandowsky, S. (2019). Addressing the theory crisis in psychology. *Psychonomic Bulletin & Review*, *26*(5), 1596–1618. https://doi.org/10.3758/s13423-019-01645-2

Palmer, C. J., Seth, A. K., & Hohwy, J. (2015). The felt presence of other minds: Predictive processing, counterfactual predictions, and mentalising in autism. *Consciousness and Cognition*, *36*, 376–389. https://doi.org/10.1016/j.concog.2015.04.007

Palmeri, T. J. (1997). Exemplar similarity and the development of automaticity. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *23*(2), 324.

Palminteri, S., Wyart, V., & Koechlin, E. (2017). The Importance of Falsification in Computational Cognitive Modeling. *Trends in Cognitive Sciences*, *21*(6), 425–433. https://doi.org/10.1016/j.tics.2017.03.011

Peebles, D., & Cooper, R. P. (2015). Thirty Years After Marr's Vision: Levels of Analysis in Cognitive Science. *Topics in Cognitive Science*, *7*(2), 187–190. https://doi.org/10.1111/tops.12137

Popper, K. R. (1959). *The Logic of Scientific Discovery*. Routledge.

Pulvermüller, F., Tomasello, R., Henningsen-Schomers, M. R., & Wennekers, T. (2021). Biological constraints on neural network models of cognitive function. *Nature Reviews Neuroscience*, *22*(8), 488–502. https://doi.org/10.1038/s41583-021-00473-5

Purves, D. (2021). *Why Brains Don't Compute*. Springer Nature.

R Core Team. (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. https://www.R-project.org/

Rauss, K., & Pourtois, G. (2013). What is Bottom-Up and What is Top-Down in Predictive Coding? *Frontiers in Psychology*, *4*. https://doi.org/10.3389/fpsyg.2013.00276

Rescorla, M. (2020). The Computational Theory of Mind. In E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy* (Fall 2020). Metaphysics Research Lab, Stanford University. https://plato.stanford.edu/archives/fall2020/entries/computational-mind/
(Date of Access: 22.1.2022)

Robert, C. (2007). *The Bayesian choice: From decision-theoretic foundations to computational implementation*. Springer Science & Business Media.

Robert, C. P., Elvira, V., Tawn, N., & Wu, C. (2018). Accelerating MCMC algorithms. *WIREs Computational Statistics*, *10*(5), e1435. https://doi.org/10.1002/wics.1435

RStudio Team. (2020). *RStudio: Integrated Development Environment for R*. RStudio, PBC. http://www.rstudio.com/

Samuels, R. (2019). Classical computational models. In M. Sprevak & M. Colombo (Eds.), *The Routledge Handbook of the Computational Mind* (pp. 103–119). Routledge, Taylor & Francis Group.

Samuelson, P. A. (1937). A note on measurement of utility. *The Review of Economic Studies*, *4*(2), 155–161.

Schulz, E., & Dayan, P. (2020). Computational Psychiatry for Computers. *IScience*, *23*(12), 101772. https://doi.org/10.1016/j.isci.2020.101772

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, *6*(2), 461–464.

Seth, A. K. (2013). Interoceptive inference, emotion, and the embodied self. *Trends in Cognitive Sciences*, *17*(11), 565–573. https://doi.org/10.1016/j.tics.2013.09.007

Shiffrin, R. M., & Nobel, P. A. (1997). The art of model development and testing. *Behavior Research Methods, Instruments, & Computers*, *29*(1), 6–14.

Smolensky, P. (1990). Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, *46*(1–2), 159–216.

Stafford, T. (2012). How do we use computational models of cognitive processes? In *Connectionist models of neurocognition and emergent behavior: From theory to applications* (pp. 326–342). World Scientific.

Stephen, I., & Sulikowski, D. (2019). *Tinbergen's Four Questions*. https://doi.org/10.1007/978-3-319-16999-6_1347-1

Sun, R. (2007). The importance of cognitive architectures: An analysis based on CLARION. *Journal of Experimental & Theoretical Artificial Intelligence*, *19*(2), 159–193.

Tauber, S., Navarro, D. J., Perfors, A., & Steyvers, M. (2017). Bayesian models of cognition revisited: Setting optimality aside and letting data drive psychological theory. *Psychological Review*, *124*(4), 410–441. https://doi.org/10.1037/rev0000052

Tenenbaum, J. B., Kemp, C., Griffiths, T. L., & Goodman, N. D. (2011). How to Grow a Mind: Statistics, Structure, and Abstraction. *Science*, *331*(6022), 1279–1285. https://doi.org/10.1126/science.1192788

Van den Bos, W., & McClure, S. M. (2013). Towards a general model of temporal discounting. *Journal of the Experimental Analysis of Behavior*, *99*(1), 58–73.

van Ravenzwaaij, D., Cassey, P., & Brown, S. D. (2018). A simple introduction to Markov Chain Monte–Carlo sampling. *Psychonomic Bulletin & Review*, *25*(1), 143–154. https://doi.org/10.3758/s13423-016-1015-8

VandenBos, G. R. (2007). *APA dictionary of psychology*. American Psychological Association.

Vehtari, A., Gelman, A., & Gabry, J. (2017). Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing*, *27*(5), 1413–1432. https://doi.org/10.1007/s11222-016-9696-4

Vernon, D. (2014). *Artificial Cognitive Systems: A Primer*. MIT Press.

Vul, E., Goodman, N., Griffiths, T. L., & Tenenbaum, J. B. (2014). One and Done? Optimal Decisions From Very Few Samples. *Cognitive Science*, *38*(4), 599–637. https://doi.org/10.1111/cogs.12101

Wagenmakers, E.-J. (2007). A practical solution to the pervasive problems ofp values. *Psychonomic Bulletin & Review*, *14*(5), 779–804. https://doi.org/10.3758/BF03194105

Wagenmakers, E.-J., Verhagen, J., & Ly, A. (2016). How to quantify the evidence for the absence of a correlation. *Behavior Research Methods, 48*(2), 413–426.

Wasserman, L. (2013). *All of statistics: A concise course in statistical inference*. Springer Science & Business Media.

Wiehler, A., Petzschner, F. H., Stephan, K. E., & Peters, J. (2017). Episodic Tags Enhance Striatal Valuation Signals during Temporal Discounting in pathological Gamblers. *ENeuro*, *4*(3). https://doi.org/10.1523/ENEURO.0159-17.2017

Wiggins, B. J., & Chrisopherson, C. D. (2019). The replication crisis in psychology: An overview for theoretical and philosophical psychology. *Journal of Theoretical and Philosophical Psychology*, *39*(4), 202.

Wixted, J. T. (2007). Dual-process theory and signal-detection theory of recognition memory. *Psychological Review*, *114*(1), 152.

Yonelinas, A. P., & Parks, C. M. (2007). Receiver operating characteristics (ROCs) in recognition memory: A review. *Psychological Bulletin*, *133*(5), 800.

Zhang, L., & Gläscher, J. (2020). A brain network supporting social influences in human decision-making. *Science Advances*, *6*(34), eabb4159. https://doi.org/10.1126/sciadv.abb4159

7 Appendix

7.1 Delay discounting code

7.1.1 Main R file – Delay Discounting Example

```
#
===================================================================
==========

# Main R file to execute Stan scripts for intertemporal choice task

# Master thesis MEi Cognitive Science

# Student: Fabian Renz (fabian.renz@univie.ac.at)

# Supversion: Lei Zhang & Paolo Petta


#
===================================================================
==========

#### Construct Data ####

#
===================================================================
==========

# clear workspace

rm(list=ls(all=TRUE))

library(rstan)

library(loo)

library(ggplot2)

library(shinystan)


#### read raw ------------------------------------------------------------------


rawdata = read.delim('data/raw_data.txt')

# rawdata = read.delim(file.choose(), header = TRUE, sep = "", dec = ".")

colnames(rawdata)[7]= "choice"

rawdata = na.omit(rawdata)


#### Preprocess the data -------------------------------------------------------

subjList  = unique(rawdata[,"subjID"])
```

```
nSubjects = length(subjList)

Tsubj = as.vector( rep( 0, nSubjects ) ) # number of valid trials per subj

for ( s in 1:nSubjects )  {
   curSubj  = subjList[ s ]
   Tsubj[s] = sum( rawdata$subjID == curSubj )
}

maxTrials = max(Tsubj)
delay_later   = array(0, c(nSubjects, maxTrials) )
amount_later  = array(0, c(nSubjects, maxTrials) )
delay_sooner  = array(0, c(nSubjects, maxTrials) )
amount_sooner = array(0, c(nSubjects, maxTrials) )
choice = array(0, c(nSubjects, maxTrials) )

for (s in 1:nSubjects) {
   curSubj      = subjList[s]
   useTrials    = Tsubj[s]
   tmp          = subset(rawdata, rawdata$subjID == curSubj)
   delay_later[s, 1:useTrials]  = tmp$delay_later
   amount_later[s, 1:useTrials] = tmp$amount_later
   delay_sooner[s, 1:useTrials]  = tmp$delay_sooner
   amount_sooner[s, 1:useTrials] = tmp$amount_sooner
   choice[s, 1:useTrials] = tmp$choice
}

dataList = list(
   nSubjects = nSubjects,
   nTrials   = maxTrials,
   Tsubj     = Tsubj,
   choice    = choice,
   amount_later   = amount_later,
    delay_later    = delay_later,
    amount_sooner  = amount_sooner,
    delay_sooner   = delay_sooner
)


#
====================================================================
==========
#### Running Stan ####
#
====================================================================
==========
rstan_options(auto_write = TRUE)
options(mc.cores = 8)

nIter     = 2000
nChains   = 4
nWarmup   = floor(nIter/2)
nThin     = 1

#### run the hyperbolic model ----------------------------------------
modelFile1 = 'scripts/hyperbolic_corrected_completed.stan'

cat("Estimating", modelFile1, "model... \n")
startTime = Sys.time(); print(startTime)
cat("Calling", nChains, "simulations in Stan... \n")

fit_hyperbolic = stan(modelFile1,
                  data    = dataList,
                  chains  = nChains,
                  iter    = nIter,
                  warmup  = nWarmup,
                  thin    = nThin,
               init    = "random",
               seed    = 145015634)

cat("Finishing", modelFile1, "model simulation ... \n")
endTime = Sys.time(); print(endTime)
cat("It took",as.character.Date(endTime - startTime), "\n")

#### run the simple heuristic model --------------------------------------
modelFile2 = 'scripts/heuristic_completed.stan'

cat("Estimating", modelFile2, "model... \n")
startTime = Sys.time(); print(startTime)
cat("Calling", nChains, "simulations in Stan... \n")

fit_heuristic = stan(modelFile2,
              data    = dataList,
              chains  = nChains,
              iter    = nIter,
              warmup  = nWarmup,
              thin    = nThin,
              init    = "random",
              seed    = 145015634)

cat("Finishing", modelFile2, "model simulation ... \n")
endTime = Sys.time(); print(endTime)
cat("It took",as.character.Date(endTime - startTime), "\n")



#### run the delay discounting exponential model --------------------------------------
modelFile3 = 'scripts/dd_exp.stan'

cat("Estimating", modelFile3, "model... \n")
startTime = Sys.time(); print(startTime)
```

```
cat("Calling", nChains, "simulations in Stan... \n")

fit_dd_exp = stan(modelFile3,
                  data    = dataList,
                  chains  = nChains,
                  iter    = nIter,
                  warmup  = nWarmup,
                  thin    = nThin,
                  init    = "random",
                  seed    = 145015634)

cat("Finishing", modelFile3, "model simulation ... \n")
endTime = Sys.time(); print(endTime)
cat("It took",as.character.Date(endTime - startTime), "\n")

#
======================================================================
==========
#### Model selection ####
#
======================================================================
==========
LL_hyperbolic = extract_log_lik(fit_hyperbolic)
LL_heuristic  = extract_log_lik(fit_heuristic)
LL_dd_exp = extract_log_lik(fit_dd_exp)

waic_hyperbolic = waic(LL_hyperbolic)
waic_heuristic  = waic(LL_heuristic)
waic_dd_exp  = waic(LL_dd_exp)

loo_hyp = loo(LL_hyperbolic)
loo_heu  = loo(LL_heuristic)
loo_dd_exp  = loo(LL_dd_exp)
```

```
# Model weights

loo_model_weights(list(LL_hyperbolic, LL_heuristic, LL_dd_exp)) # probability of the model
to be the winning one


#
===================================================================
==========

#### Model inspection ####

#
===================================================================
==========


# Posterior Analysis for winning model - extract group level parameter (hyperbolic model)

# slope of the sigmoid - large beta indicates non random/deterministic choices

mu_beta_plot = stan_plot(fit_hyperbolic, pars = c("mu_beta"), show_density=TRUE,
ci_level=.95, outer_level=.99)

# deprication factor - the larger the quicker subjective value depricates

mu_k_plot = stan_plot(fit_hyperbolic, pars = c("mu_k"), show_density=TRUE, ci_level=.95,
outer_level=.99)


launch_shinystan(fit_hyperbolic)

print(fit_hyperbolic)

posterior <- extract(fit_hyperbolic)


#### End of file
```

7.1.2 Heuristic model – Ericson (2015)

```
// Heuristic model (Ericson, 2015)
// STAN file Master thesis MEi Cognitive Science
// Student: Fabian Renz (fabian.renz@univie.ac.at)
// Supversion: Lei Zhang & Paolo Petta

data {
    int<lower=1> nSubjects;
    int<lower=1> nTrials;
    int<lower=1, upper=nTrials> Tsubj[nSubjects];
    real<lower=0> delay_later[nSubjects,nTrials];
    real<lower=0> amount_later[nSubjects,nTrials];
    real<lower=0> delay_sooner[nSubjects,nTrials];
    real<lower=0> amount_sooner[nSubjects,nTrials];
    int<lower=0,upper=1> choice[nSubjects, nTrials]; // 0 for instant reward, 1 for delayed reward

}

parameters {
  // Hyper(group)-parameters for hierachical model
  vector[5] mu;
  vector<lower = 0> [5] sigma;

  // Subject-level raw parameters
  vector[nSubjects] beta_Int_0;   // just created structure, no filling yet
  vector[nSubjects] beta_AmAb_0; // Amount Absolute
  vector[nSubjects] beta_AmRe_0; // Amount Relative
  vector[nSubjects] beta_DeAb_0; // Delay Absolute
  vector[nSubjects] beta_DeRe_0; // Delay Relative
}

transformed parameters {
  // Transform subject-level raw parameters
  vector[nSubjects] beta_Int_1;   // just created structure, no filling yet - subject level parameters
  vector[nSubjects] beta_AmAb_1;
  vector[nSubjects] beta_AmRe_1;
  vector[nSubjects] beta_DeAb_1;
  vector[nSubjects] beta_DeRe_1;

  beta_Int_1  = mu[1] + sigma[1] * beta_Int_0;  // just created structure, no filling yet, linking the two together
  beta_AmAb_1 = mu[2] + sigma[2] * beta_AmAb_0; // hierachical model mu & sigma group level parameters
  beta_AmRe_1 = mu[3] + sigma[3] * beta_AmRe_0;
  beta_DeAb_1 = mu[4] + sigma[4] * beta_DeAb_0;
  beta_DeRe_1 = mu[5] + sigma[5] * beta_DeRe_0;
}

model {
  // Hyperparameters

  mu        ~ normal(0, 1);  // priors group level parameters
  sigma     ~ cauchy(0, 3);

  // individual parameters - priors individual level
  beta_Int_0   ~ normal(0, 1);   // we could also use cauchy, if we want it to be super uninformative, we could use (0,10)
  beta_AmAb_0 ~ normal(0, 1);
  beta_AmRe_0 ~ normal(0, 1);
  beta_DeAb_0 ~ normal(0, 1);
  beta_DeRe_0 ~ normal(0, 1);

  for (s in 1:nSubjects) {
    // Define values
    real G;
    real R;
    real D;
    real T;

    for (t in 1:(Tsubj[s])) {
      // creating each term individually
      G = amount_later[s,t] - amount_sooner[s,t];
      R = (amount_later[s,t] - amount_sooner[s,t]) / ((amount_later[s,t] + amount_sooner[s,t])/2);
      D = delay_later[s,t] - delay_sooner[s,t];
      T = (delay_later[s,t] - delay_sooner[s,t]) / ((delay_later[s,t] + delay_sooner[s,t])/2);
      // bernoulli logit is logistic regression function --> either smaller soon or larger later option
      choice[s,t] ~ bernoulli_logit( beta_Int_1[s] + beta_AmAb_1[s] * G + beta_AmRe_1[s] * R + beta_DeAb_1[s] * D + beta_DeRe_1[s] * T );
    }
  }
}

generated quantities {

  real mu_beta_Int;
  real mu_beta_AmAb;
  real mu_beta_AmRe;
  real mu_beta_DeAb;
  real mu_beta_DeRe;

  real log_lik[nSubjects];

  mu_beta_Int  = mu[1];  // we didn't specify before which is which in the vector in the model block, so here we declare it
  mu_beta_AmAb = mu[2];
  mu_beta_AmRe = mu[3];
  mu_beta_DeAb = mu[4];
  mu_beta_DeRe = mu[5];
  { // local section, this saves time and space
    for (s in 1:nSubjects) {
      real G;
      real R;
      real D;
      real T;

      log_lik[s] = 0;



      for (t in 1:(Tsubj[s])) {

      G = amount_later[s,t] - amount_sooner[s,t];
      R = (amount_later[s,t] - amount_sooner[s,t]) / ((amount_later[s,t] + amount_sooner[s,t])/2);
      D = delay_later[s,t] - delay_sooner[s,t];
      T = (delay_later[s,t] - delay_sooner[s,t]) / ((delay_later[s,t] + delay_sooner[s,t])/2);

      // lpmpf = log probability mass function - adds on each iteration; log to deal with small values otherwise running into a numerical problem
      log_lik[s] += bernoulli_logit_lpmf( choice[s,t] | beta_Int_1[s] + beta_AmAb_1[s] * G + beta_AmRe_1[s] * R + beta_DeAb_1[s] * D + beta_DeRe_1[s] * T );



      }
    }
  }
}
```

7.1.3 Hyperbolic model – Mazur (1987)

```
// Hyperbolic model (Mazur, 1987)
// STAN file Master thesis MEi Cognitive Science
// Student: Fabian Renz (fabian.renz@univie.ac.at)
// Supversion: Lei Zhang & Paolo Petta

data {
    int<lower=1> nSubjects;
    int<lower=1> nTrials;
    int<lower=1, upper=nTrials> Tsubj[nSubjects];
    real<lower=0> delay_later[nSubjects,nTrials];
    real<lower=0> amount_later[nSubjects,nTrials];
    real<lower=0> delay_sooner[nSubjects,nTrials];
    real<lower=0> amount_sooner[nSubjects,nTrials];
    int<lower=0,upper=1> choice[nSubjects, nTrials]; // 0 for instant reward, 1 for delayed reward

    //int<lower=0> N_mis;
}

parameters {
  // Hyper(group)-parameters
  real mu_k_raw; // group parameters mu - k is deprication factor
  real mu_beta_raw; // beta indicates the slope of the sigmoid/softmax
  real<lower = 0> sd_k_raw;
  real<lower = 0> sd_beta_raw;

  //real y_mis[N_mis];

  // Subject-level raw parameters
  vector[nSubjects] k_raw;
  vector[nSubjects] beta_raw;
}
```

```
transformed parameters {
  // Transform subject-level raw parameters
  vector<lower=0,upper=1>[nSubjects] k; // deprication of value
  vector<lower=0,upper=5>[nSubjects] beta; // beta higher slope of softmax --> more deterministic


  for (s in 1:nSubjects) {


    k[s]    = Phi_approx( mu_k_raw + sd_k_raw * k_raw[s] ); // constraining the parameters
    beta[s] = Phi_approx( mu_beta_raw + sd_beta_raw * beta_raw[s] ) * 5;


  }
}


model {
  // Hyperparameters
  mu_k_raw    ~ normal(0, 1); // priors for parameters
  mu_beta_raw ~ normal(0, 1);
  sd_k_raw    ~ cauchy(0, 3);
  sd_beta_raw ~ cauchy(0, 3);


  // individual parameters
  k_raw    ~ normal(0, 1);
  beta_raw ~ normal(0, 1);


  for (s in 1:nSubjects) {
    // Define values
    real ev_later;
    real ev_sooner;


    for (t in 1:(Tsubj[s])) {
      ev_later   = amount_later[s,t]  / ( 1 + k[s] * delay_later[s,t] );
      ev_sooner  = amount_sooner[s,t] / ( 1 + k[s] * delay_sooner[s,t] );
      // ernoulli logit is logistic regression function --> either smaller soon or larger later option
      choice[s,t] ~ bernoulli_logit( beta[s] * (ev_later - ev_sooner) ); // beta determines the slope of the sigmoid
    }
  }
}

generated quantities {
  real<lower=0,upper=1> mu_k;
  real<lower=0,upper=5> mu_beta;
  real log_lik[nSubjects];

  mu_k    = Phi_approx(mu_k_raw);
  mu_beta = Phi_approx(mu_beta_raw) * 5;

  { // local section, this saves time and space
    for (s in 1:nSubjects) {
      real ev_later;
      real ev_sooner;

      log_lik[s] = 0;

      for (t in 1:(Tsubj[s])) {
        ev_later   = amount_later[s,t]  / ( 1 + k[s] * delay_later[s,t] );
        ev_sooner  = amount_sooner[s,t] / ( 1 + k[s] * delay_sooner[s,t] );
        // lpmpf = log probability mass function - adds on each iteration; log to deal with small values otherwise running into a numerical problem
        log_lik[s] += bernoulli_logit_lpmf( choice[s,t] | beta[s] * (ev_later - ev_sooner) );
      }
    }
  }
}
```

7.1.4 Exponential model – Samuelson (1937)

```
//Exponential delay discounting model (Samuelson, 1937)
// STAN file Master thesis MEi Cognitive Science
// Student: Fabian Renz (fabian.renz@univie.ac.at)
// Supversion: Lei Zhang & Paolo Petta

data {
   int<lower=1> nSubjects;
   int<lower=1> nTrials;
   int<lower=1, upper=nTrials> Tsubj[nSubjects];
   real<lower=0> delay_later[nSubjects,nTrials];
   real<lower=0> amount_later[nSubjects,nTrials];
   real<lower=0> delay_sooner[nSubjects,nTrials];
   real<lower=0> amount_sooner[nSubjects,nTrials];
   int<lower=0,upper=1> choice[nSubjects, nTrials]; // 0 for instant reward, 1 for delayed reward
}

parameters {
  // Hyper(group)-parameters
  real mu_r_raw;
  real mu_beta_raw;
  real<lower=0> sd_r_raw;
  real<lower=0> sd_beta_raw;

  // Subject-level raw parameters
  vector[nSubjects] r_raw;
  vector[nSubjects] beta_raw;
}

transformed parameters {
  // Transform subject-level raw parameters
  vector<lower=0,upper=1>[nSubjects] r;
  vector<lower=0,upper=5>[nSubjects] beta;

  for (s in 1:nSubjects) {
    r[s]    = Phi_approx( mu_r_raw + sd_r_raw * r_raw[s] );
    beta[s] = Phi_approx( mu_beta_raw + sd_beta_raw * beta_raw[s] ) * 5;
  }
}

model {
  // Hyperparameters
  mu_r_raw    ~ normal(0, 1);
  mu_beta_raw ~ normal(0, 1);
  sd_r_raw    ~ cauchy(0, 3);
  sd_beta_raw ~ cauchy(0, 3);

  // individual parameters
  r_raw    ~ normal(0, 1);
  beta_raw ~ normal(0, 1);

  for (s in 1:nSubjects) {
    // Define values
    real ev_later;
    real ev_sooner;

    for (t in 1:(Tsubj[s])) {
      ev_later  = amount_later[s,t]  * exp( -1*r[s]*delay_later[s,t] );
      ev_sooner = amount_sooner[s,t] * exp( -1*r[s]*delay_sooner[s,t] );
      // Bernoulli logit is logistic regression function --> either smaller soon or larger later option
      choice[s,t] ~ bernoulli_logit( beta[s] * (ev_later - ev_sooner) ); // beta determines the slope of the sigmoid
    }
  }
}
```

```
generated quantities {
  real<lower=0,upper=1> mu_r;
  real<lower=0,upper=5> mu_beta;

  real log_lik[nSubjects];

  mu_r    = Phi_approx(mu_r_raw); // r acts like the k in the hyperbolic model
  mu_beta = Phi_approx(mu_beta_raw) * 5;

  { // local section, this saves time and space
    for (s in 1:nSubjects) {
      // Define values
      real ev_later;
      real ev_sooner;

      log_lik[s] = 0;

      for (t in 1:(Tsubj[s])) {
        ev_later  = amount_later[s,t]  * exp( -1*r[s]*delay_later[s,t] ); // only difference compared to hyperbolic model - here exponential term
        ev_sooner = amount_sooner[s,t] * exp( -1*r[s]*delay_sooner[s,t] ); // -1 so the exponent is negative and value decreases over time
        // lpmpf = log probability mass function - adds on each iteration; log to deal with small values otherwise running into a numerical problem
        log_lik[s] = log_lik[s] + bernoulli_logit_lpmf( choice[s,t] | beta[s] * (ev_later - ev_sooner) );
      }
    }
  }
}
```

7.2 Cauchy distribution

```
# -*- coding: utf-8 -*-
"""
Cauchy distribution illustration

Author: Fabian Renz (fabian.renz@univie.ac.at)
Master thesis University of Vienna MEi: CogSci
"""

import numpy as np
from scipy.stats import cauchy
from matplotlib import pyplot as plt



#------------------------------------------------------------
# Define the distribution parameters to be plotted
gamma_values = [1.0, 3.0, 5.0]
mu = 0
x = np.linspace(-10, 10, 1000)
color = ['r', 'b', 'g']

#------------------------------------------------------------
# plot the distributions
fig, ax = plt.subplots(figsize=(5, 3.75))

for gamma, color in zip(gamma_values, color):
    dist = cauchy(mu, gamma)

    plt.plot(x, dist.pdf(x), color=color,
             label=r'$\mu=%i,\ \gamma=%.1f$' % (mu, gamma))

plt.xlim(-4.5, 4.5)
plt.ylim(0, 0.65)
plt.xlabel('$\\theta$')
plt.ylabel(r'$p(\theta|\mu,\gamma)$')
plt.title('Cauchy Distribution')

plt.legend()
plt.show()

# save figure
fig.savefig('../Plots/Cauchy_illustrtaion.png', dpi=500)
```

7.3 Abstract

Modelling is a commonly employed tool in cognitive science to approach phenomena too complex or too difficult to deal with directly. This thesis addresses the use of computational models in cognitive science. The first part gives an overview across the different uses of computational models as well as important distinctions in computational cognitive modelling. These include the difference between descriptive models, capturing empirical regularities and cognitive process models, theorizing about the cognitive mechanisms underlying observed behaviour. Furthermore, different taxonomies and David Marr’s levels of explanation are introduced. The second part of the thesis covers Bayesian methods in cognitive modelling as one example subfield, introducing different applications of Bayes’ theorem along a case study from delay discounting research, illustrating the key concepts in Bayesian computational modelling applied to 3 models. In conclusion the limitations and criticisms of both Bayesian and other computational cognitive models are discussed as well as their offerings to cognitive science. This thesis is meant for everyone interested in the role of computational modelling in cognitive science and scopes different applications of computational models. Furthermore, it introduces the reader to Bayesian approaches and walks through the application of Bayesian cognitive models applied to a concrete example. Thereby, tying the conceptual ideas to a concrete example demonstrating the offerings and limitations to the reader to hopefully make them more accessible.

7.4 Deutsche Zusammenfassung

Die Modellierung ist ein häufig verwendetes Werkzeug in den Kognitionswissenschaften. Sie ermöglicht die Auseinandersetzung mit Phänomenen, welche zu komplex sind, um sie direkt in ihrer Gänze zu behandeln. Diese Thesis beschäftigt sich mit computationaler Modellierung in den Kognitionswissenschaften. Der erste Teil gibt einen Überblick über verschiedene Verwendungen von computationlaer Modellierung, und führt grundlegende Unterscheidungen ein. Dazu gehören deskriptive Modelle, welche empirische Regularitäten beschreiben und kognitive Prozessmodelle. Letztere stellen Theorien dar über die zugrundeliegenden kognitiven Prozesse menschlichen Verhaltens. Im Anschluss werden verschiedene Taxonomien eingeführt sowie "bottom-up" und "top-down" Ansätze. Im Zweiten Teil der Thesis werden bayesianische Methoden in der kognitiven Modellierung behandelt als ein Beispielfeld der computationalen kognitiven Modellierung. Dabei liegt der Fokus auf verschiedenen Anwendungsbereichen des Bayes' Theorem gefolgt von einer Fallstudie aus dem Forschungsfeld „Delay Discounting". Delay Discounting beschreibt den subjektiven Verfall von Wert aufgrund zeitlicher Verzögerung. Im Rahmen der Fallstudie werden die wichtigsten Konzepte der bayesianischen kognitiven Modellierung demonstriert und aktiv auf 3 Modelle angewendet. Im dritten und letzten Teil der Thesis werden die Limitationen von (bayesianischer) Computermodellierung diskutiert und deren Beiträge zu den Kognitionswissenschaften.

Diese Thesis ist an alle gerichtet, die sich für die Rolle computationaler Modellierung interessieren und illustriert verschiedene Anwendungen computationaler Modellierung in den Kognitionswissenschaften. Die Fallstudie demonstriert aktiv die Verwendung bayesianischer Methoden, um die eingeführten Konzepte zugänglich zu machen, sowie Möglichkeiten und Limitationen computationaler kognitiver Modellierung aufzuzeigen.

7.5 Acknowledgements

I would like to thank Anne for supporting me throughout the past years. Furthermore, I would like to point out the contributions of Lei Zhang, who was not allowed to be listed as a co-supervisor. However, he had a tremendous influence on this thesis and helped me understand how to perform modelling in practice and supported me in the transition towards a PhD - thank you! And lastly, I would like to thank Paolo who not only supervised me during this thesis, but throughout the entire program and beyond. Thank you for teaching me how to approach science, ask questions and more broadly how to think about cognition.