REINFORCEMENT LEARNING APPROACHES TO THE ANALYSIS OF THE EMERGENCE OF GOAL-DIRECTED BEHAVIOUR

CONSTANTINOS MITSOPOULOS

Department of Psychological Sciences

Centre for Brain and Cognitive Development

Centre for Cognition, Computation and Modelling

This dissertation is submitted for the degree of

Doctor of Philosophy

Birkbeck, University of London

2016

To my father who encouraged me to widen my horizons
To my mother, grandmother and brother for their immense love and support
To my uncle who inspired me to become a scientist

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other University. This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration, except where specifically indicated in the text.

CONSTANTINOS MITSOPOULOS
2016

Acknowledgements

Throughout the four happy and fruitful years I spent in Birkbeck, many contributed to this work in various ways. First and foremost, I would like to acknowledge the tireless and prompt help of my advisor Rick Cooper. His constant support, guidance and very detailed feedback helped me to shape my ideas into actual research projects and accomplish this work. Equally, I would like to thank my second advisor Denis Mareschal for introducing me to the Developmental Psychology, for all our stimulating discussions and for his unique ability to get me back on track whenever I was at a 'research dead-end'. I am grateful to both, for giving me complete freedom to define and explore my own research directions and provide me with anything I needed for my research (... and for their patience!).

I feel privileged having as an "unofficial" advisor Peter Dayan. I am grateful for all the time he spent into our meetings, teaching me the latest advances in Reinforcement Learning, answering to my many-times-naïve questions and shaping my scientific thinking. I also thank him for letting me be part of the Gatsby "family" and join all the inspiring talks there.

I feel greatly indebted to my friend Arthur Guez. I thank him for guiding me since my initial explorations in the Reinforcement Learning field, introducing me to adaptive planning and all the exciting research that comes with it. I thank him for all his advises and discussions we had throughout all these years.

I really thank Sabine Hunnius, Claire Monroy and Ezgi Kayhan for making my stay in Nijmegen so pleasant and smooth. I feel grateful for having the chance to collaborate with them in two different projects. I feel that I need to especially thank Claire Monroe for spending so much time reading the whole thesis and giving me invaluable feedback. I feel grateful also to Bert Kappen for our many-hour meetings that led me to more theoretical paths of the Reinforcement Learning field, introducing me to the stochastic optimal control theory and many other exciting topics.

During the four years-programme I had the chance to meet, interact and collab-

orate with other researchers that helped me in different ways: Livia Freier, Zlatko Franjcic, Katharina Kaduk, Oscar Portolés Marín, Estefanía Domínguez Martínez, Áine Ni Choisdealbha, Janna Gottwald and Benjamin Koch. I thank also the European Commission for financial support (grant MC-ITN-289404-ACT).

Coming from a completely mathematical field, interacting with psychologists brings lots of challenges on the table. One of this is the art of presentation. I am happy that now I can easily communicate my research to people with non technical background, a skill that proved extremely useful in my life. For this, and for many other things, I thank Massimo Stocci. You changed my life forever!

I thank Stefania Sguera for all her support, encouragement love and patience throughout the last two years of my studies. Finally, I would like to thank my family. All these years your support and love was an invaluable ally that help me to move forward and face any difficulties. I will be forever grateful. I thank also my uncle and my grandmother, who left this world happy knowing that this work is completed. To all those who believed in me and supported me I hope this thesis makes you proud.

Abstract

Over recent decades, theoretical neuroscience, helped by computational methods such as Reinforcement Learning (RL), has provided detailed descriptions of the psychology and neurobiology of decision-making. RL has provided many insights into the mechanisms underlying decision-making processes from neuronal to behavioral levels. In this work, we attempt to demonstrate the effectiveness of RL methods in explaining behavior in a normative setting through three main case studies.

Evidence from literature shows that, apart from the commonly discussed cognitive search process, that governs the solution procedure of a planning task, there is an online perceptual process that directs the action selection towards moves that appear more 'natural' at a given configuration of a task. These two processes can be partially dissociated through developmental studies, with perceptual processes apparently more dominant in the planning of younger children, prior to the maturation of executive functions required for the control of search. Therefore, we present a formalization of planning processes to account for perceptual features of the task, and relate it to human data.

Although young children are able to demonstrate their preferences by using physical actions, infants are restricted because of their as-yet-undeveloped motor skills. Eye-tracking methods have been employed to tackle this difficulty. Exploring different model-free RL algorithms and their possible cognitive realizations in decision making, in a second case study, we demonstrate behavioral signatures of decision making processes in eye-movement data and provide a potential framework for integrating eye-movement patterns with behavioral patterns.

Finally, in a third project we examine how uncertainty in choices might guide exploration in 10-year-olds, using an abstract RL-based mathematical model. Throughout, aspects of action selection are seen as emerging from the RL computational framework. We, thus, conclude that computational descriptions of the developing decision making functions provide one plausible avenue by which to normatively

characterize and define the functions that control action selection.

Contents

List of Figures

List of Tables

Chapter 1

Computational Approaches to Decision Making and Behavioral Analysis

ABSTRACT

This chapter introduces the two main formal methods of investigation used throughout this thesis: reinforcement learning and Bayesian model fitting. The chapter provides preliminary discussion of the purpose and requirements of each technique, and closes with a summary of the goals of the thesis and its structure.

1.1 Introduction

Over the last decade, an explosive growth in the use of Machine Learning methods within the behavioral and cognitive sciences has occurred. This is arguably for the reason that Machine Learning approaches include mathematical models that describe various types and aspects of learning processes. Therefore, they form a suitable overall framework to describe various behavioral and cognitive phenomena.

Machine Learning methods can be classified into three main categories based on the type of learning that is involved: Supervised Learning, Reinforcement Learning and Unsupervised Learning. In this thesis we will focus on Reinforcement Learning models in order to demonstrate their use and strengths in modeling decision making processes. The suggested models will be fit to human data by an approach known as Bayesian inference. This aims to find, for a parametrized model and a dataset, the values of the parameters that maximize the probability of the data given the

Method	Problems	Examples
Supervised Learning	Prediction	predict next day's temperature
Reinforcement Learning	Optimization	select actions to maximize profit
Unsupervised Learning	Structure of Data	find similarities among data

Table 1.1: Machine Learning methods overview. The three main types of learning can be used to learn to solve different types of problems. Supervised Learning is used to train models in order to learn to make predictions given inputs. Reinforcement Learning trains models in order to be able to map inputs to actions that ultimately will lead to the maximization of a specific utility function. Unsupervised Learning is used in order to discover structural relationships among the input data or create compact representations of the latter (e.g., image compression).

model. We begin by presenting a concise overview of the three types of learning (a high level overview with examples is presented in table 1.1).

Supervised Learning consists of the family of models that learn the functional relationship of a system between its input variables and the observed responses. The main problem related to Supervised Learning techniques is prediction of the target response from input data. During the training phase of this type of model, the data used contain examples of inputs and outputs collected by actually measuring the system (or the process) that is being modeled. Furthermore at each training episode the model outputs a response which is compared to the actual observed response and the difference of the two is used to update the parameters of the model. This kind of training is named supervised training or training with a teacher as the model is presented with a response for each input sample, and a 'teacher' or supervisor provides either the correct answer or an error related to how close/far the model's response falls from the actual system's response. This is simply characterized by a loss/error function which is specified by the model designer.

Unsupervised Learning techniques, or learning without a teacher, are mainly models that capture structure or identify properties of the data. A hidden structure can be captured if data are projected into a lower or higher dimensional space, thus creating lower or higher representations of the data. By this process, additional information such as groupings among data points that might be formed can be identified. These representations can be very useful as they can effectively compress information and extract only the most important features (dimensions) of it. Additionally one might be interested in identifying similarities among the data

items. Thus, unsupervised learning methods are mainly used for dimensionality reduction and clustering. Unsupervised learning methods lack a direct measure of success, such as the loss function in supervised learning methods. Thus, it is difficult to assess the validity of inferences drawn from the output of such techniques. One must resort to graphical visualizations of the results or heuristics that could possibly provide a measure of effectiveness of the method used, as the latter is a matter of opinion and cannot be objectively verified.

Reinforcement Learning is the field of Machine Learning that studies decision making processes and in general can be seen as an optimization method. The main framework consists of an agent whose goal is to find the best actions in an environment in order to maximize a utility function. At each time step the agent takes an action which affects the state of the environment and receives a reward regarding how good or bad that action was. The reward can be sparse and delayed which makes the problem difficult as the agent cannot map easily every course of action to a specific value of the reward. Furthermore, the agent has to explore sufficiently in order to discover behaviors that lead to better outcomes.

One can see Reinforcement Learning methods as lying in between Supervised and Unsupervised Learning methods. Naturally, someone could wonder about the differences between the supervised learning methods described above. We illustrate these differences with a simple example. We assume that we have built a robot and we would like to program it in order to drive a car. In a supervised learning fashion we would provide to the robot many pair-examples of images of the road and proper actions related to the steering of the wheel. The robot will try to minimize the error between its actions and the correct actions provided by us. Hopefully, the robot will learn which angle it should steer the wheel in order to drive according to the samples provided to it. If the training examples were collected by a bad driver the robot will adopt similar behavior. In a Reinforcement Learning context, we can design a general reward function (e.g., penalizing the robot when it hits other cars or gets out of its lane) and let the robot decide which actions are good or bad given the images of the road. In this case the robot will try to optimize the reward function by exploring different action sequences and eventually will learn an optimal behavior under reward constraints.

In this thesis we demonstrate the application of the Machine Learning framework in Developmental Cognitive Neuroscience. More specifically, we examine the effectiveness of Reinforcement Learning methods for describing and explaining the

development of various decision making processes in humans, such as planning and learning to decide by trial and error. As discussed above, Reinforcement Learning is suitable in scenarios in which an agent has to take decisions in order to maximize a reward function or achieve a goal. Thus, in the experimental design the reward is explicitly expressed either in the form of accumulated points or a signal/gesture that something important was achieved. Furthermore, we attempt to characterize the behavioral patterns of different aged population by specific model parameters.

Our main methodological approach is Bayesian model fitting of Reinforcement Learning models on behavioral data collected from various experiments. The following two sections describe in broad terms the elements of the two formal approaches (Reinforcement Learning and Bayesian model fitting) used mainly throughout the whole thesis, while the final section of this chapter outlines the remaining chapters of this thesis.

1.2 Elements of Reinforcement Learning

As discussed above, Reinforcement Learning (RL) is a behavior-inspired type of learning focused on the problem of a learning agent (human, animal, robot etc) that tries to achieve a goal by interacting with its surrounding environment. The core idea behind RL is learning by obtaining rewards, while avoiding punishments, for selecting actions in order to achieve a goal. A block diagram that illustrates the agent-environment interaction is given in fig. 1.1. Below, we provide some informal definitions of the key elements of RL in fig. 1.1.

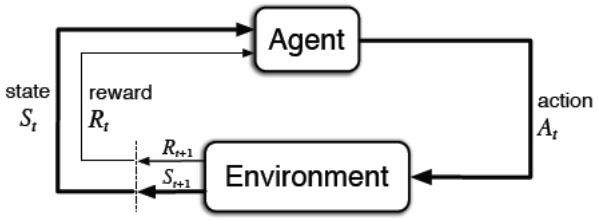


Figure 1.1: Agent-Environment interaction.

- **Agent:** The agent is the learner or decision-maker. It might be an animal, a human, a robot or a machine. In general, an agent can perceive, act and learn.

- **Environment:** Everything that the agent interacts with can be considered part of the environment. The agent, by using its sensors or sensory organs, perceives the state of the environment and selects an action. The environment changes in response to these actions and presents new situations to the agent. The environment can also contain a reward that the agent can receive and attempt to maximize over time.

- **States:** The set of environmental states $\mathcal{S}$, represents the unique characterization of all the important information regarding the environment. This set can be finite or infinite, and discrete or continuous. For example, the configuration of a chessboard is a state in the game of chess. All states in a chess game form a large but finite and discrete set of states. To give another example, a particle that moves in space can be characterized by its coordinates (relative to the origin of a coordinate system) and velocity. These characteristics, at every time step, could also be considered as a form of state (i.e. $s_t = (\mathbf{x}, \dot{\mathbf{x}})$, where the first vector element is the particle's position vector and the second one the particle's velocity vector). In this formulation, each element of the state vector represents a *feature* of the state. In this example, each feature can take infinite different values, and the combination of these values is also infinite; thus the state space is continuous and the number of states is infinite.

- **Action:** An action is a specific behavior performed by an agent in order to bring itself closer to a goal, which usually is to maximize the total amount of reward the agent receives in the long term. The set of actions $\mathcal{A}$ consists of all possible actions in a given domain. As with the the case of states, the action space can be discrete or continuous and finite or infinite. For example, when we have to decide which of two slot machines[1] to play, there are only two possible actions, $|\mathcal{A}| = 2$, so the action space is finite and discrete. On the other hand if we navigate with a joystick, the actions are infinite and the action space continuous, as there are infinite positions of the joystick. Furthermore, in some states, the whole set of actions A might not be available,

[1]Slot machines, also known as one-armed bandits, are gambling machines with three or more reels which spin when a button/lever is pressed/pulled. A gambler presses the button and the reels start spinning. Then pressing again the button, the reels stop and the payoff received is based on patterns of symbols visible on the reels. For example, a high payoff is usually received when the symbols on the reels, at the point where they stopped spinning, are the same.

as might be the case for an infant who does not yet have the motor ability to select some actions. The set of available actions in a particular state s is usually denoted by $\mathcal{A}(s) \subseteq \mathcal{A}$. Action selection depends on the *policy* (defined next). An important issue with regard to action selection is the trade-off between *exploration* and *exploitation*. The term exploration refers to the tendency for an agent to select actions randomly, in order to test the outcomes of novel options that it has never chosen before. Exploration enables the agent to sample experience from the environment. When the agent makes use of this experience, and selects actions according to it, he or she can exploit the acquired knowledge. Exploitation is the tendency to limit one's action choices to only those which have been selected previously or for which the agent already has knowledge about. In this way, the agent can make use of previous knowledge about the consequences of the action. The way in which a balance of exploration and exploitation is achieved, either in human behavior or an artificial agent's behavior, is an open question in the Artificial Intelligence (AI) community.

- **Policy:** The agent implements a mapping from states to actions. This mapping is called *a policy* denoted by $\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, which is a probability distribution over actions given states. The probability of taking action a at time t, at state s, $P(a_t = a|s_t = s)$ is denoted by $\pi(s, a)$. It can be deterministic (i.e using specified criteria to select an action in a particular state) or stochastic (i.e using a probability distribution over actions given states). One common choice is to use the Boltzmann distribution, which describes the probability of selecting an action given the "energy"[2] $E(s, a)$ of that action and the value of the exploration/exploitation parameter or reward sensitivity β:

$$\pi(a_t|s_t) = \frac{e^{\beta E(s_t,a_t)}}{\sum_{a'} e^{\beta E(s_t,a')}} \tag{1.1}$$

where a' denotes all possible actions at the given state. Alternatively, the ϵ-greedy or ϵ-soft greedy policy is designed such that for a fixed probability

[2]Here we use the term "energy" to resemble the usage of this function with its usage in Statistical Mechanics were it originated. Later this "energy" will be substituted by value functions which will be introduced later in this chapter.

ϵ the agent explores and for $1-\epsilon$ exploits:

$$\pi(a_t|s_t) = \begin{cases} \epsilon & \text{, choose random action} \\ 1-\epsilon & \text{, choose action that max } E(s,a) \end{cases} \tag{1.2}$$

Using the above 'glossary' the RL problem can be framed as follows: Agent and environment interact at each of a sequence of discrete time steps $t = 0, 1, 2, \ldots$[3] and at each temporal instance the agent receives some representation of the environment's state $s_t \in \mathcal{S}$ and selects an action, $\alpha_t \in \mathcal{A}$, where $\mathcal{S}$ and $\mathcal{A}$ are the state space and action space respectively. At time $t+1$ the agent receives a reward $r_{t+1} \in \mathbb{R}$, as a consequence of its action and then finds itself in a new environment state s_{t+1}. Reinforcement learning methods formalize how the agent changes its policy as a result of its experience in order to maximize the total amount of reward received.

1.3 Bayesian Model Fitting

Scientific modeling consists of making hypothesis about the nature of the underlying function of a process. One aspect of modeling makes use of explicit mathematical expressions that aim to describe the relationship between a stimulus and the observations that are evoked by it. Many times this mapping is of great interest but the form of the underlying function is unknown, or it is not derived by a physical law and thus does not have an explicit expression. One way to overcome this difficulty and approximate the underlying process is to use computational approaches.

Some of the computational approaches attempt to capture (or learn) the general behavioral pattern of the function that describes the generative process of our data, especially when that process is rather complicated and cannot be expressed by an explicit formula. In order to use these methods we need to make an initial assumption about the function that maps stimuli to observations. For this reason we characterize the input-output relationship as a parametrized function of the input. In order to approximate the whole process and make predictions, we need

[3] Time is considered discrete for simplicity. In addition, for the sequence to converge, there are two formulations which we need to consider. The first one, in which the time horizon is infinite and the reward is discounted by a factor γ; and the second one, in which the time horizon is finite, for example $t = 0, 1, 2, ...T$, and no discount factor. For generality we consider the infinite case with discount factor γ.

to estimate the parameters that under the initial hypothesis, reflected in the form of the function selected, lead to the best fit of the observed data.

In general, we can consider two possible avenues for fitting a model to a collected data set. One method is to update the parameters of the model in a way that the error between actual response and model's response gets minimized. Another approach is to consider that the observed data were generated by a model which is represented as a parametrized probability distribution. Therefore, the response is treated as a random variable. In this approach, we attempt to find the parameters of the probability distribution (probability density function) that maximize the likelihood of the observed data. If we have prior knowledge over the possible parameters, then using Bayes rule we can estimate a distribution over possible parameters given the data, and attempt to maximize this quantity. Thus, we will end up having a distribution over model parameters which best describes the responses. In both cases we can use the resulting models to make predictions for unseen stimulus. In case the observed responses are normally distributed, the two approaches are equivalent (see Bishop (2006) for a detailed derivation), although in general the two approaches tend to differ. In this thesis, we adopt the probabilistic method to estimate the parameters of a model.

The main assumption in Bayesian modeling is that the parameters of the model are not fixed, rather they are sampled from a probability distribution (the prior). As we described, someone can use a likelihood function, that describes the observed data, along with a prior distribution over the parameters of that likelihood function in order to get a probability distribution over the parameters given the data (the posterior). The posterior distribution not only enables us to make predictions from new input data, but also it can provide a measure of how uncertain we are for that prediction. This is one of the main advantages of Bayesian methods. Furthermore, in some cases in which there is available knowledge regarding the possible range of values of the parameters, the usage of prior distributions over parameters can help to restrict the optimal parameters search to that range of values leading to better parameter estimates.

For example, the outcome of tossing a coin can be described by a random variable that takes values 0 or 1 representing head or tail respectively. The likelihood of getting a specific outcome can be described by a Bernoulli distribution with parameter p representing the probability of the coin to appear one of the two outcomes after being tossed. Therefore, we express the likelihood of the coin resulting

in Heads (or Tails) as a parametrized distribution function (a model). After a number of tosses we can use the collected data and the model in order to predict how likely it is for our toss to result in head. For this we need to estimate what is the value of the model parameter that most likely generated the collected data. In order to estimate the model parameter we can use maximum likelihood techniques.

Although maximum likelihood methods provide a good estimate of the model parameter, there might be cases in which prior knowledge about the distribution or the range of the values of the parameter is available. Furthermore, we might not be interested in a point estimate of the model parameter, rather we may be interested in estimating a probability distribution over all possible values of it. Hence, we can make use of Bayes rule (likelihood $\times$ prior) and estimate the posterior distribution of the model parameter. This estimation can be implemented by sampling the resulting distribution of the product of the likelihood and the prior distributions, or using again a point estimation (maximum a posteriori) to find the most likely model parameter that maximizes the posterior distribution given the observed data. Eventually we will end up not only with a model parameter estimation that is most likely responsible for generating our data, but also with an estimation of the uncertainty of how accurate our calculation is.

From the aforementioned approach it is apparent that we assume a form of stochasticity in the model parameters' nature. As we will see in the following chapters, human behavior appears to have a form of noise and we wish this to be reflected by our modeling approach. For this reason, our models will assign a probability to any possible action. This immediately creates a form of a probabilistic model of action and accounts for the stochastic nature of human or animal actions. Thus, the determination of a model parameter given any observed data can be quantified statistically under the notion of the probability it assigns to the data. Therefore, Bayesian Inference is a reasonable and mathematically justified choice for fitting the models to their data.

1.4 Organization of the Thesis

Each chapter of the main body of this thesis describes a separate case study that addresses questions regarding decision making problems, mainly in children compared to adults. The modeling efforts presented provide a mechanistic description of behavior in a normative setting. Normative behavior emerges from optimal ac-

tion selection — optimal in a Reinforcement Learning sense. Organisms are viewed as decision makers that attempt to select their actions in order to maximize their expected reward, given (or not) knowledge about the environment's action-reward structure and constraints on the kinds of computations they can perform. Optimality is expressed explicitly or implicitly in all case studies in the form of a reward signal that is present during, or at the end, of each task.

The work presented is separated into three main case studies:

- The first one is driven by questions regarding mental planning. Various tasks were used in order to explore different aspects of mental planning, and the modeling approach was fixed across tasks.
- In the second study we show how learning action-effect relationships can be viewed as a reward maximization problem. We used behavioral results of one experimental procedure but propose various theoretical and computational interpretations.
- The third study focuses on interpreting the behavioral patterns of one experimental task using only one mathematical model with different components that have cognitive interpretations. The reward signal in the task follows every single action choice of the participant.

Where it is feasible, attempts are made to link the modeling components to functional aspects of cognition.

In each case study chapter we first introduce the reader to the scientific literature regarding the context of the problems we address. Next, we present the computational formulation (according to Reinforcement Learning framework) of these problems and then we describe the behavioral experimental procedures and subsequent data analysis. Finally, we discuss the lessons that can be learned by applying the RL framework and Bayesian model fitting to the specific type of problem.

The detailed structure of this thesis is as follows:

Chapter 2: This chapter provides a more detailed introduction to the RL framework, key theoretical and computational results, and discusses briefly cognitive applications of it. It introduces the reader to the MDP (Markov Decision Processes) framework and the traditional algorithmic solutions of them. We also briefly discuss model-based and model-free

RL and their links to goal-directed and habitual behavior respectively. We then demonstrate some of the RL algorithms with a simple toy example, in which an artificial agent learns to navigate in a room domain in order to get to a specific spatial location. Finally, we provide a review of the neurological and behavioral evidence of RL and discuss how the method has been used to characterize the patterns of striatal dopamine and action selection in organisms.

Chapter 3: This chapter gives a detailed description of the Bayesian methods we use to fit RL models. This links the algorithms described in the previous chapter with actual behavior. We illustrate these with a simple example simulating a situation in which someone has to decide to gamble between two slot machines.

Chapter 4: This chapter presents a case study of action selection in mental planning tasks in which the reward is very sparse (i.e., given only on completion of the task). Step-by-step motivation is introduced as a reward shaping function (Ng et al. (1999)) and is used to guide RL tree search models that account for planning. To examine the phenomenon, we use two different planning tasks to collect human behavioral data. Furthermore, we compare the fitted models' parameters across different age groups and consider cognitive interpretations of parameters within models.

Chapter 5: The main goal of the case study presented in this chapter is to model computationally the process that humans utilize in order to learn from a demonstration video, and then transfer this knowledge to the real world. It was conducted with collaboration with the Radbound Institute of Nijmegen university in Netherlands, as part of the student exchange policy of the Marie Curie early-stage researcher programme. The specific project was co-supervised by Sabine Hunnius and the research was carried out along with Claire Monroy, in a duration of 3 months. We use various RL models and formalization of the problem along with eye-tracking and behavioral data to explore the learning mechanism that takes place during the experiment. Finally, we compare adults performance with infants' performance and extract insights on their respectful behavior.

Chapter 6: This case study explores the relation of exploration with uncertainty on available options. The work was also conducted in Radbound Institute with collaboration with Ezgi Kayhan. The author's contribution was mainly on the computational modeling and data analysis parts.

Chapter 7: In this chapter we conclude the thesis. We highlight the main contributions and stress the usefulness of Reinforcement Learning models in explaining the behavioral patterns for each case study. Furthermore, we analyze the limitations of the models used and what could be done to potentially overcome these. A section in which future directions are discussed, is provided at the end of the chapter.

All the computational work was implemented by the author of this thesis without the use of any external decision making toolboxes (unless it is explicitly stated). In addition, the experiments that are represented in a computerized form were also designed and programmed by the author.

Chapter 2

Reinforcement Learning I: Concepts and Algorithms

ABSTRACT

In this chapter we introduce the mathematical foundations of Reinforcement Learning methods. We present in detail the characteristics of two main RL categories, Model-based and Model-free, and how these methods can be algorithmically realized. Apart from the different examples of RL algorithms, we illustrate their application with a simple navigation problem at the end of the chapter. This demonstrates how RL agents can generate optimal behavior in a given environment. Additionally, we describe how RL relates to cognitive neuroscience by explaining the firing of dopaminergic neurons and linked neuronal and behavioral scales.

2.1 Introduction

Humans and animals are required to make decisions at each step in their daily life, from muscle movements to higher cognitive decisions such as deciding between options that will change the course of their future (e.g., education, investments, family, etc.). Some of these decisions are consciously considered and processed whereas others, such as muscle movements, are not. Some decisions are the result of a long learning process beginning during early developmental stages, such as the coordination of the limbs.

Decision theory, which can be applied in different fields (economics, mathematics, neuroscience, psychology, etc.) and levels (from deciding what clothes to wear for a particular occasion to coordination of dance movements), potentially provides a unifying framework for investigating action selection. Fundamental questions

about how such a decision-making system develops, remain unanswered.

Most of the time, action decisions involve a situation in which an agent comes across a state of the world (the environment in which the agent exists and interacts) in which it needs to select an action in order to achieve a goal, complete a task, or cause a change to that initial state. Decision theory quantifies what action should be chosen in the context of a given utility function (defined below) and some knowledge of the environmental states. The decision-theoretic framework combines a probabilistic model of the world with the goals of an agent that are formalized by a utility function. Using this framework it is possible to make predictions on how someone would decide within a specific context.

A utility function is a hypothesized function that increases with increasing desirability of the outcome. Usually, it is assumed that an agent tries to maximize the expected utility which is defined as the sum over all products between the probability of being in a particular state and the reward at this state (the value of the utility function at this state). By choosing according to this criterion the agent selects its actions rationally. To give a simple example, we can imagine how someone could decide how to maximize his or her profit by gambling with two slot machines.

In general, decision theory relies on three parts. The first establishes a probabilistic relationship between an agent's actions and the states of the world that the agent may achieve. The second quantifies the value of being in possible future states, in terms of expected reward from these states, and the third combines these components formulating an optimization problem to select the optimal choice or option.

One issue that needs to be taken into account is the uncertainty of the state of the environment that the agent inhabits. In most realistic situations the agent has partial knowledge about the real state of the environment. However, given some observations that the agent's sensory system might receive, it is possible to assign a probability distribution over the real states of the world, and thus attempt to make a decision with only the partial information that is available. An illustrative example is given by Cassandra et al. (1994):

> "*You stand in front of two doors: behind one door is a tiger and behind the other is a vast reward, but you do not know which is where. You may open either door, receiving a large penalty if you choose the one with the tiger and a large reward if you choose the other. You have the*

> *additional option of simply listening. If the tiger is on the left, then with probability 0.85 you will hear it on your left and with probability 0.15 you will hear it on your right; symmetrically for the case in which the tiger is on your right. If you listen you will pay a small penalty. Immediately after you choose either of the doors and receive the reward or the big penalty in case the tiger was behind the door, the problem resets and you will again be faced with the two doors choice with the tiger randomly repositioned. How long should you stand and listen before you choose a door?*" Cassandra et al. (1994, p. 7)

In the above problem, it is clear how important is the action selection. The term action selection can be interpreted in various ways. For example, action selection could refer to motor control commands, or to high-level cognitive tasks such as making career-related decisions. Throughout this thesis, we conceptualize action selection as the process of choosing the next action during each step in a sequential task. It combines habitual/routine sequential behaviour and intentional goal-directed behaviour.

Following Norman and Shallice (1986), performance of an action may be automatic or under deliberate control. An automatic action is performed without the agent who executes it being aware of an intention to perform it, such as a reflex. A general situation of this type is in the initiation of routine actions or habitual behaviour. Such behaviour is usually elicited by a given stimulus. The specific response is the one that is most strongly associated with the stimulus. The stimulus comes from the environment in which the agent acts.

The control of action though is more complicated. Actions that can be classified as automatic can also often be carried out under deliberate conscious control when desired. These action sequences have the ability to run themselves automatically, without conscious control, yet to be modulated by conscious control when needed. Possibly these phenomena occur by the interaction between two different functions: one that will account for intentional control of action (top down control) and a second one that will be initiated by environmental stimuli (bottom up, affordances).

According to Gibson's concept of affordances (Gibson, 1986), the sight of an object triggers a sensorimotor process that generates neural activity relating to the most suitable grasp for that object based on the contextual information, but a single object may activate different affordance representations in the anterior intraparietal (AIP) area of the brain. This bottom-up, sensory-driven activation of

multiple affordances is automatic and independent of the requirements of the task at hand, according to evidence from psychophysical experiments (Caligiore et al., 2010).

At the same time, in a particular task reward-driven behavior can lead to a suppression of stimulus-driven behavior. Thus, the interplay between top-down and bottom-up control of action is of great importance but it complicates the decision-making problem. From a developmental perspective, three factors need to be taken into account: Top-down intentions, bottom-up affordances, and the emergence of habits.

The necessity to integrate the three sources of information, which develop along different time scales, is apparent. Mathematical and computational approaches offer potential insight into how these factors might interact, and result in successful control of action. One critical question, that we attempt to address in our work, is the role of learning in the above context: how do humans or animals learn their preferences for different actions and outcomes?

Reinforcement Learning is a computational framework that combines learning and decision making, as an agent learns its preferences by interaction with the environment. This framework has been used successfully to interpret empirical results from behavioral experiments that involve action selection problems in order to complete sequential tasks (for extensive reviews refer to Dayan and Daw (2008); Lee et al. (2012); Niv (2009)). Unlike abstract decisions analyzed in economic theories, biological organisms do not always receive information about the consequences of selecting alternative actions. Instead, they face the challenge of learning how to predict the outcomes of their actions by trial and error, which is the core of reinforcement learning.

Our intentions are to investigate how action selection develops through infancy and childhood, using computational approaches from Machine Learning (and in particular Reinforcement Learning) and mathematical methods from Bayesian Statistics. Using these, we will explore how internal models for action selection may emerge and be combined or modulated by intention, and how such a process may be represented in the developing cognitive system. This will lead to a better understanding of how such a decision making system for action selection may develop, and how this system is influenced by the environment.

One can distinguish between habits and goal-directed actions (Dickinson, 1985; Wood and Neal, 2007). Reinforcement Learning has been linked to goal-directed as

well as habitual behaviour (Balleine and Dickinson, 1998). More specifically, new knowledge about the environment can be acquired without experiencing directly any reward, perhaps by inferring the outcome of a future state using a model as a representation of the dynamics of the environment that an agent could maintain. This is referred to as model-based RL. It contrasts with model-free RL, where the agent relies entirely on experienced rewards and penalties. For example, a model-free architecture such as Actor-Critic (described in section 2.4), can store preferences in the form of artificial synaptic weights between state and actions which represent stimulus-response associations. More specifically:

Model-based: With the term "model" we do not mean anything else than a distribution over transitions $\mathcal{P}(s, a, s')$(i.e., probabilities of moving from state s to state s' using action a) and a distribution over rewards $\mathcal{R}(s, a, s')$(i.e., probability of receiving a reward r following the transition). This is a very generic concept; if the environment (or the reward delivery system) is deterministic, then the notion of distribution is not appropriate and the case is reduced to the so-called tabular case, where one can store one value per state in an array allocated in a computer's main memory (Szepesvári, 2010). A model can be used in various ways to find optimal solutions using different algorithms that are categorized within Dynamic Programming methods. With respect to cognitive or behavioral concepts, RL models are considered formal descriptions of how an agent plans ahead its actions.

Model-free: These types of methods use no models of the environment in order to learn. The term Reinforcement Learning is usually linked only to model-free methods, whereas model-based methods are linked to Dynamic Programming. A main difference between model-based and model-free learning is that the agent needs to sample enough knowledge from the environment in order achieve its goal, and usually this is characterized as trial-and-error learning. To illustrate the difference we give a simple example in which an agent with a very basic model-based system faces a very basic model-free agent: an agent that has some knowledge about the rules of chess, and another agent that has no knowledge of chess. The first agent can plan, which means it can think ahead about its moves and their consequences, evaluate them, and then act. The second agent, however, will start moving the pieces, without following any rules, and upon receiving feedback (from an imaginary

referee) will attempt to correct its actions to be consistent with the game rules. Its moves will be completely random at the beginning, and only later will it learn that some moves are good and some are bad. The other agent will use its model to evaluate different possible outcomes from its current state, including searching some steps ahead. This gives the model-based agent the advantage of choosing actions which will most likely have rewarding outcomes. On the other hand, a disadvantage is that an inaccurate model will lead to non-rewarding actions.

Model-based and model-free algorithms are very important aspects of decision making. Especially in psychology, researchers have argued for years that the human brain uses both of these systems, when making decisions (see Dolan and Dayan 2013 for a review). We clarify here that the computational approaches of these two systems are also useful when modeling human behavioral data.

2.2 Mathematical Formulation

Sequential decision making problems, such as the ones mentioned in the previous sections (e.g., chess), can be described by the Markov Decision Processes (MDPs) framework. Before defining an MDP we will first describe the concept of Markov chains. A Markov chain is a model for a random process that evolves over time, such that the next state is dependent on the current state but not prior states. A simple infinite Markov chain is given in fig. 2.1 which represents the transition probabilities from state to state. The property described above is referred to as *Markov property* given formally by:

$$P(s_{t+1}|s_1, \cdots, s_t) = P(s_{t+1}|s_t) \tag{2.1}$$

This property is very important for memoryless procedures, as it describes how an agent is able to plan and act using only the information available its current state. To illustrate this we can use the example of a chess game in which a player can plan a move even with his/her chess pieces already set in an arbitrary configuration. All the necessary information for planning and acting is given in the current state.

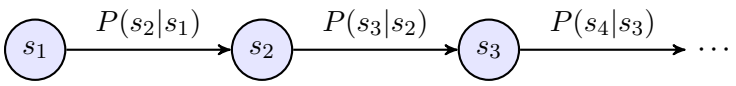


Figure 2.1: A simple Markov chain (first order Markov Model). Each state depends only on its previous state (Markov property.

2.2.1 Markov Decision Processes (MDPs)

An MDP (Puterman, 1994) is a model for controlled random processes in which an agent's choice is determined by the probabilities of transitions within a Markov chain and leads to rewards. Formally, a MDP is a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$ where $\mathcal{S}$ is a finite set of states, $\mathcal{A}$ a finite set of actions, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ a state transition probability matrix with elements $p^a_{ss'} = \mathcal{P}(s'|s, a)$, indicating the probability of transition from state s to s' by selecting action a, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ a reward function and $\gamma \in [0, 1]$ a discount factor.

When all the components of the tuple are known, standard *Dynamic Programming* (Bertsekas, 2000) algorithms can be used to obtain the optimal value function. We define the total discounted reward at time-step t:

$$v_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{2.2}$$

A reward received k time-steps in the future is worth only γ^{k-1} times what it would be worth if it were received immediately. Values of γ close to 0 lead to "myopic" evaluation (consider only current rewards) whereas values close to 1 lead to "far-sighted" evaluation, meaning that the learner values long term rewards almost as much as immediate rewards. It is worth noting that eq. 2.2 has an infinite *horizon* but with the discount factor it is guaranteed to converge. Another alternative would be to use a finite horizon, in which case we could have the choice to omit the discount factor.

We introduce here two very important value functions[1], the *state-value function* and the *action-value function*:

- **State-Value function** of an MDP is defined as the expected return starting

[1]Value functions are functions that give an estimate of the total reward expected in future, starting from each state and following a particular policy

from state s, and then following policy π:

$$V^{\pi}(s) = \mathbb{E}_{\pi}[v_t | s_t = s] \tag{2.3}$$

This gives an estimate of how good it is for the agent to be in a given state.

- **Action-Value function** of an MDP is defined as the expected return starting from state s, taking action a, and then following policy π:

$$Q^{\pi}(s,a) = \mathbb{E}_{\pi}[v_t | s_t = s, a_t = a] \tag{2.4}$$

This gives an estimate of how good it is for the agent to perform an action in a given state.

2.2.2 Bellman Equations

The Value functions described above utilize specific recursive equations that help to compute them, for any state s and policy π. These equations are called Bellman Equations (Bellman, 1954) and are given by:

$$V^{\pi}(s) = \sum_{a \in \mathcal{A}} \pi(s,a) \sum_{s' \in \mathcal{S}} \mathcal{P}^{a}_{ss'} \left(\mathcal{R}^{a}_{s} + \gamma V^{\pi}(s') \right) \tag{2.5}$$

$$Q^{\pi}(s,a) = \mathcal{R}^{a}_{s} + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}^{a}_{ss'} \sum_{a' \in \mathcal{A}} \pi(s',a') Q^{\pi}(s',a') \tag{2.6}$$

These equations can be easily derived if we decompose the corresponding value function (eq. 2.3, 2.4) into the sum of immediate reward and discounted reward of the successor state (for detailed derivation the reader is referred to Sutton and Barto (1998, p.70)).

2.2.3 Bellman Optimaility Equation

The goal for any given MDP is to find the policy that receives the most reward. To find the optimum policy we need to maximize one of the equations 2.5,2.6. By definition, the optimum value functions are given by:

$$V^{*}(s) = \max_{\pi} \{ V^{\pi}(s) \} \tag{2.7}$$

$$Q^*(s,a) = \max_{\pi} \{Q^{\pi}(s,a)\} \tag{2.8}$$

The optimal value functions specify the best possible performance in an MDP, and this MDP is considered solved if these functions are known. For any MDP, there exists an optimal policy that is better or equal to all other policies. All optimal policies achieve the optimal state/action-value function. An optimal policy can be found by maximizing $Q^*(s,a)$. Then the optimal state-value function is related to the optimal action-value function by:

$$V^*(s) = \max_{a} \{Q^*(s,a)\} \tag{2.9}$$

According to the above and eq. 2.9, and using the decomposition of the value function to immediate reward and successor state, the *Bellman Optimality Equation* is given by

$$V^*(s) = \max_{a} \left\{ \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a V^*(s') \right\} \tag{2.10}$$

with

$$Q^*(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a V^*(s') \tag{2.11}$$

To find the *optimal policy* a general rule can be applied:

$$\pi^* = \arg\max_{a} \{Q^*(s,a)\} \tag{2.12}$$

The Bellman Optimality Equation expresses the fact that the value of a state under an optimal policy must equal the expected return for the best action from that state. For any MDP there is at least one optimal deterministic policy.

2.2.4 Partially Observable MDPs (POMDPs)

There are occasions where the agent cannot determine the current state with complete reliability. An appropriate way to model sequential decision making processes under such uncertainty is to formalize these kinds of problems into POMDPs. Using such a model an agent can plan optimally according to its belief by taking into account the uncertainty associated with its actions and observations.

A POMDP is generally defined as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \Omega, \mathcal{O}, \gamma)$ where:

- $\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma$ describe an MDP;

- Ω is a finite set of observations the agent can experience in the enviornment;
- $\mathcal{O} : \mathcal{S} \times \mathcal{A} \to \Pi(\Omega)$ is the observation function which gives a probability distribution over possible observations for each action and resulting state.

An important difference between MDPs and POMDPs is that in the case of POMDPs the state of the world is hidden, and the only information that the agent has is an observation that it receives at each state. Thus the agent maintains and updates a probability distribution over the states of the partially observable environment which represents its belief about the state of the world. In short, an agent being in belief state b performs action a and receives an observation o. Then, it updates its belief state according to $b' = \tau(b, a, o)$ which is the probability of doing the above transition. The agent can update its current belief state b according to Bayes rule, using the belief update function:

$$b'(s') = const \cdot \mathcal{O}(s', a, z) \sum_{s \in \mathcal{S}} \mathcal{P}(s, a, s') b(s) \tag{2.13}$$

where *const* is the normalization constant.

The solution of a POMDP consists in finding an optimal policy π^* which specifies the best action choice in every belief state and depends on the planning horizon and the discount factor. In order to do that we need to compute the optimal value of a belief state over the planning horizon which, for infinite horizon, is given:

$$V^*(b) = \max_{a \in \mathcal{A}} \left\{ \mathcal{R}(b, a) + \gamma \sum_{o \in \Omega} \mathcal{P}(o|b, a) V^*(\tau(b, a, o)) \right\} \tag{2.14}$$

Then the optimal policy is defined as:

$$\pi^*(b) = \arg\max_{a \in \mathcal{A}} \left\{ \mathcal{R}(b, a) + \gamma \sum_{o \in \Omega} \mathcal{P}(o|b, a) V^*(\tau(b, a, o)) \right\} \tag{2.15}$$

The function $b(s)$ is defined into a continuous space, and thus can take infinite values. This allows an infinite number of belief states, making it intractable to compute a policy for all possible belief states in a finite amount of time. However, the optimal value function over a finite horizon is piecewise linear and convex (Kaelbling et al., 1998) and a very close approximation to V^* can be computed in a finite amount of time. The exact value iteration algorithm and variations of it have been used to compute optimal policies in partially observable stochastic

domains. However exact algorithms can be applied only to small problems of 10 to 20 states due to their high complexity (Littman, 1996).

Approximate value iteration algorithms are used in more complex domains such as: Point Based Value Iteration (PBVI) by Pineau et al. (2003), which bounds the complexity of exact value iteration to the number of belief points in its set; Perseus algorithm by Spaan and Vlassis (2005), which instead of updating all belief points at each iteration, updates only the belief points which have not been improved in the current iteration; HSVI (Smith and Simmons, 2004, 2005) which uses an heuristic to select the belief point on which to do value iteration updates; Other interesting strategies provide online approaches in which the policy needs to be computed only for the belief states that are encountered during the execution (Kearns et al., 2002; Paquet et al., 2005; Ross, 2007; Ross and Chaib-Draa, 2007; Satia and Lave, 1973; Washington, 1997).

2.3 MDP Solvers

Value functions are non-linear and in general there is no closed form solution; for this reason iterative solution methods are preferred. A vast variety of methods for solving the Reinforcement Learning problem exists in the scientific literature. Most of the main (value-based[2]) methods have three key ideas in common (Sutton and Barto, 1998):

- The objective of all the algorithms is the estimation of value functions.
- Their main operation is to back up values along state trajectories (i.e., gathering the reward at the end of a trajectory and propagating it back to the start of the trajectory).
- They maintain an approximate value function and an approximate policy, and they try to improve each on the basis of the other.

In general, all value-based MDP solvers tackle an MDP in two phases:

1. **Policy Evaluation:** The solver uses a fixed policy to calculate the value function for some or all of the states.

[2]All algorithms used in this thesis use value functions. Example of non-value-based algorithms are policy search algorithms.

2. **Policy Improvement:** The algorithm improves the previous policy using values obtained in the policy evaluation step.

The steps 1 and 2 are repeated until an ending criterion (e.g., the policy remains unchanged after some iterations).

Furthermore a main categorization of the methods depends on the existence or not of an MDP model (*planning* algorithms use such a model whereas *learning* algorithms do not). An illustration of the relation between the different methods can be seen in fig. 2.3. The three main method categories are:

- **Model-based:** A model-based algorithm learns a model $\mathcal{P}(s'|s,a)$ of the environment and uses this model to update the value function and derive a policy. The model might be given a priori, or learned by experience on-line, and the algorithm executes model-based control design on the estimated model. Dynamic Programming is one of the most common model-based methods used to solve RL problems, as it needs a perfect description of the environment.

- **Model-free:** It is possible to estimate value functions without using a model (defined by the distributions of transitions $\mathcal{P}$ and rewards $\mathcal{R}$). Methods that use experience, to directly learn a value function are categorized as model-free methods. Such methods are the Temporal Difference (TD), Q-Learning, SARSA, Monte Carlo methods, etc.

- **Policy Search:** Experience is used to evaluate different policies and directly search in the space of policies. Usually gradient methods are used for the search of the optimum. Some advantages of these methods are good convergence properties, effectiveness in continuous spaces and the fact that they can learn stochastic policies. The main issue in policy search methods is the computation of the gradient, which in general suffers from the common problems of gradient search methods (i.e., existence of local minima).

Another categorization of learning algorithms that should be considered, are the *On-line* and *Off-line* learning algorithms. One approach to this problem is to use a simulator for the environment and train the learner off-line. The simulator will provide many training examples and the learning procedure will be fast. This is very useful when the environment can be easily and accurately modeled (i.e., backgammon, chess, etc.), providing an advantage to the off-line learning procedure. Furthermore, because the off-line method uses simulated experience, it does

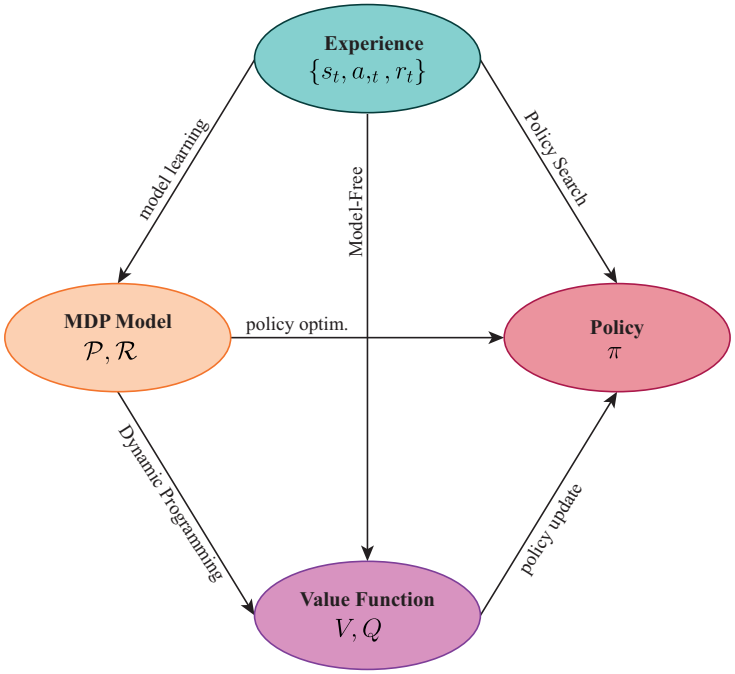


Figure 2.2: Common approaches to learning with experience: Model-free, Model-based and Policy Search. With the term Experience we mean a set of states s, actions a and rewards r. From sampled experience a model can be used (or learned) along with common Dynamic Programming approaches (e.g., Value Iteration) to estimate value functions; this consists a Model-based approach. In a Model-free approach the value functions are estimated by samples of state-action pairs along with their respective reinforcement signal. Policy search follows a different approach, in which a policy function is directly estimated from sampled experience. These approaches are basic and present a generic view of Reinforcement Learning approaches.

not face the problem of the trade-off between exploration and exploitation, as it designs the control solution before applying it to the real environment. On the other hand, when the environment can not be modeled accurately, we may wish the agent to explore it and learn. On such occasions, on-line methods are preferred.

Other methods can be characterized as *On/Off-policy* methods (e.g., off-policy Actor-Critic, off-policy TD-Learning, on-policy SARSA learning (Rummery and Niranjan, 1994). The most well-known off-policy method is Q-learning (Watkins, 1989; Watkins and Dayan, 1992). In an on-policy setting, the agent learns only about the policy it is executing. In an off-policy setting, an agent learns about a policy or policies different from the one it is executing.

2.4 A Toy Example

In this section we will describe briefly some algorithmic realizations of the various methods that solve MDPs. In order to do that, we will use a simple domain in which we can assign a task to our agent. Our domain consists of a grid map (fig. 2.3) of 169 tiles (the states) with 65 of them resembling "walls". The agent starts from the initial position, indicated in green (tile 25), and its goal is to reach the state indicated by the red color (tile 107). There it receives a reward. If an action leads it to a wall state then it remains at that position. The actions that the agent can use to reach its goal are: north, north-east, south-east, south, south-west, west, and north-west.

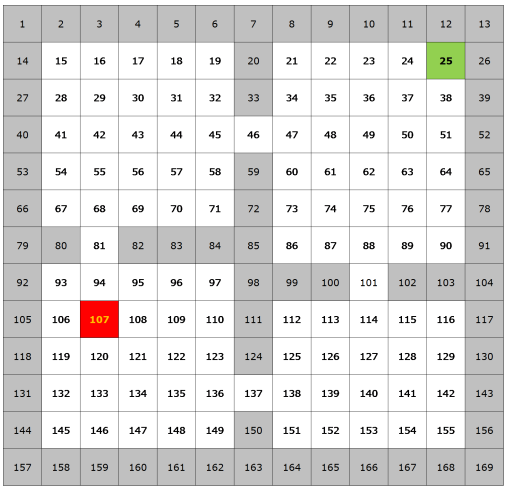


Figure 2.3: The grid world domain. Our agent will execute here a simple task: Starting from position 25 we want to end at position 107 where it gets a reward $R = 100$ and the episode terminates. The reward given at state 107 is the only reward that the agent receives in the whole domain.

The first method that we demonstrate is the model-free algorithm Q-learning, (alg. 2.1). This algorithm is characterized as model-free for the reason that it does not maintain any distribution over the dynamics (state transitions) or the rewards (model). It learns by sampling experience from the environment iteratively, initially by trying different random actions and with the received feedback it decides which actions are good or bad. Eventually it learns what the optimal actions are to select at each position in order to reach the goal state.

The second method that we use is the Actor-Critic (Barto et al., 1983) algorithm (alg. 2.2). Briefly, this algorithm explicitly represents the policy independently of

Algorithm 2.1 Q-learning. The Q function is set to either zero values or small random numbers to give to the agent some initial "preference" for actions.

Initialize $Q(s,a)$
for each episode **do**
  Initialize state s
  for each time step t of the episode **do**
    Select action a_t from $\pi(a_t|s_t)$ dependent on $Q(s_t,.)$
    Observe reward r and new state s_{t+1}
    Update:
    $Q(s_t,a_t) = Q(s_t,a_t) + \alpha[r + \gamma \max_{a'} Q(s_{t+1},a') - Q(s_t,a_t)]$
  end for
end for

Algorithm 2.2 Actor-Critic

Initialize $P(s,a)$
for each episode **do**
  Initialize state s
  for each time step t of the episode **do**
    Select action a_t from $\pi(a_t|s_t)$ dependent on $P(s_t,.)$
    Observe reward r and new state s_{t+1}
    Evaluate Temporal Difference Error:
    $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$
    Update the **actor**:
    $P(s_{t+1},a) = P(s_t,a) + \beta\delta_t$
    Update the **critic**:
    $V(s_{t+1}) = V(s_t) + \alpha\delta_t$
  end for
end for

the value function. The policy representation is known as *actor*, as it is used to select actions. The actor is evaluated by the *critic*, which typically is the value function. The evaluation is the usual temporal difference error by the critic used in the RL literature. However, the algorithm does not only update the value function but also the policy representation, indicating which action a is preferred in each state s.

Finally, we demonstrate the above algorithms in a model-based-model-free algorithmic combination, the Dyna algorithm (alg. 2.3). The Dyna algorithm creates a model of the world using experience gained by a model-free algorithm. Thus, online planning and learning from sampled experience are combined into one framework.

Given the Q function, an agent at time step t, in state s_t, decides to select an

Algorithm 2.3 The Dyna Q algorithm

Initialize $Q(s,a)$ and $Model(s,a)$
for each episode **do**
  Initialize state s
  Select action α_t observe reward r and new state s_{t+1}
  $Q(s_t,a_t) = Q(s_t,a_t) + \alpha[r + \gamma \max_{a'} Q(s_{t+1},a') - Q(s_t,a_t)]$
  $Model(s,a) \leftarrow s_{t+1}, r$
  repeat N times
    $s \leftarrow$ random previously observed state
    $a \leftarrow$ random action previously taken in s
    $s', r \leftarrow Model(s,a)$
    Update:
    $Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s_t,a_t)]$
  until
end for

Favorite model-free algorithm e.g. Q-learning.

action a_t, with probability given by the Boltzmann function (eq. 2.16)

$$p(a_t|s_t) = \frac{e^{\beta Q(s_t,a_t)}}{\sum_{a'} e^{\beta Q(s_t,a')}} \tag{2.16}$$

This is an example of the softmax policy. The agent does not always "listen" to the highest value of the Q function, but depending on β it might explore other options.

To demonstrate the performance of each of these algorithms, we predefined the number of episodes to 200 and the number of time steps of each episode to 40,000. We ran each algorithm for 100 simulations, and took the average number of time steps for each of the 200 episodes. In terms of performance Dyna algorithms need less interaction with the environment to learn the task, because they create an estimated model of the world from experience gained by the model-free methods, and for such kind of problems they perform well.

An illustration of such a performance is shown in fig. 2.4, which shows how fast an agent learns (in terms of how many steps are needed until the goal state is discovered). On average with fewer time steps needed. The Dyna architecture incorporates a planning process which is simulated with a model-free algorithm (i.e., Q-learning or Actor-Critic). Without the planning ($N = 0$) each episode adds only one step to the policy. The planning process constructs a model of the domain

which can be exploited by the agent to decide which action to select. In contrast, the model-free algorithms try to learn optimum actions by sampling experience (state transitions dependent on actions, and the resulting rewards of these actions).

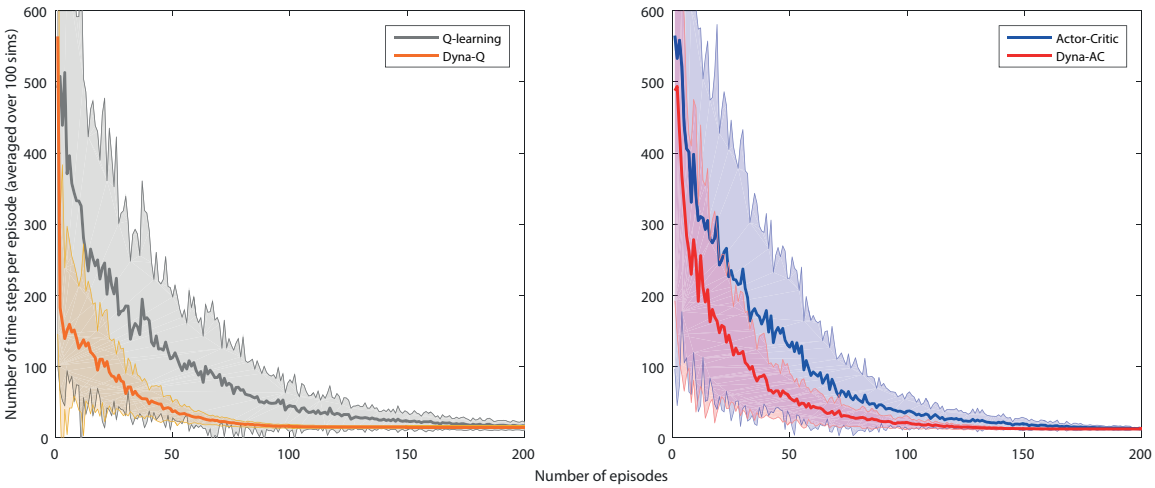


Figure 2.4: The Q-Learning, Actor Critic and the Dyna versions of each these. At the beginning, because the agent behaves randomly we observe great variance in the number of the time steps needed to complete the episode (or usually finding the goal). Later, the agents have explored sufficiently the environment, and they exploit more their knowledge so the variance drops.

The difference in those algorithms, naturally, has an effect in their collected overall reward (fig. 2.5). Because of the non-zero discount factor γ, the number of time steps affects the overall collected reward. Thus, the more time the algorithm needs to terminate an episode (i.e., finding the goal state) the less the value of the reward will be. If the agent wanders until the terminal condition of an episode, 40000 time steps, the reward will be discounted so much that it will be close to zero. As we observe, the Dyna versions of the algorithms manage to collect more reward as they reach the goal faster than their simpler counterparts.

We also present an example of the Q-value function learned throughout training (fig. 2.6). The learned Q-value function, indicates the preference of an action at a particular state. Within the figure, states with darker colors are preferred less. After the training, we can let the agent repeat the task following the policy given by eq. 2.16, and the Q function learned before. Then the agent will most likely step toward the tiles with lighter color and choose the action that is indicated by the blue arrow.

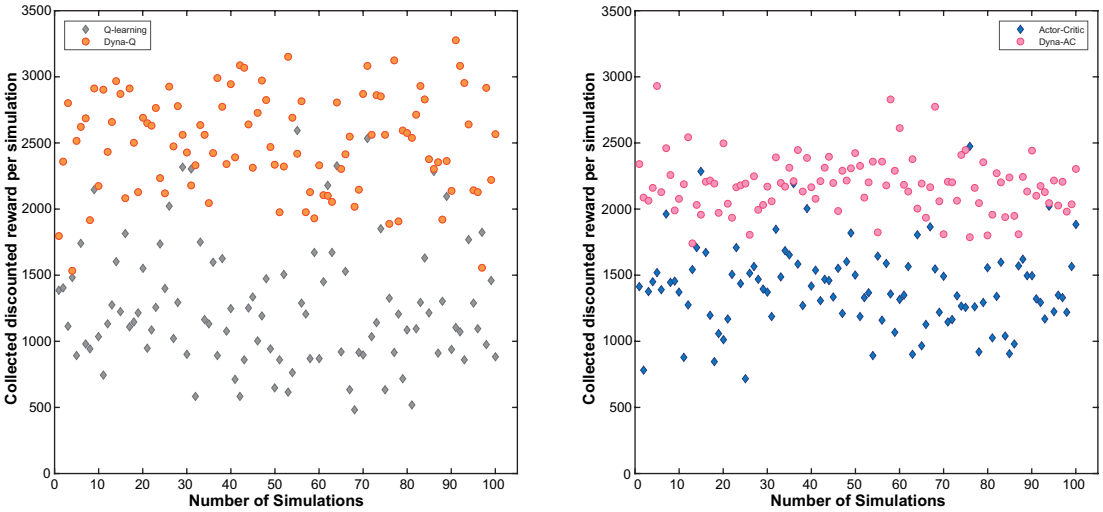


Figure 2.5: The discounted reward obtained at each simulation.

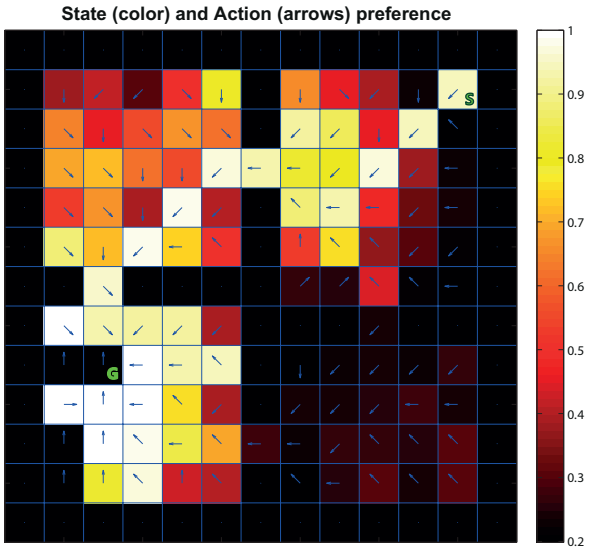


Figure 2.6: What is learned by the agent. An example of Q-value function values (normalized) after the learning process in the four rooms domain. The color gradient indicates the level of preference on a specific state (i.e., position in the room) and the arrows the corresponding preferred action (policy).

The above demonstration gives some insights into how a computational algorithm can be used to represent learning and acting in an unknown environment. Furthermore, some different architectures were discussed: a model-free approach through which the agent samples knowledge from the environment and learns, a combination of actor-and-critic architecture, and lastly we incorporated a forward model which accounts for planning.

The Actor-Critic-based model is very important because, while the model-free system gathers experience and information from the environment, the model-based planning system utilizes this information to take optimal actions. This model-free/model-based interaction will be discussed later on. The above algorithms form a basis of models that can be used to possibly explain an organism's decision-making system, and the underlying mechanisms which still remain elusive.

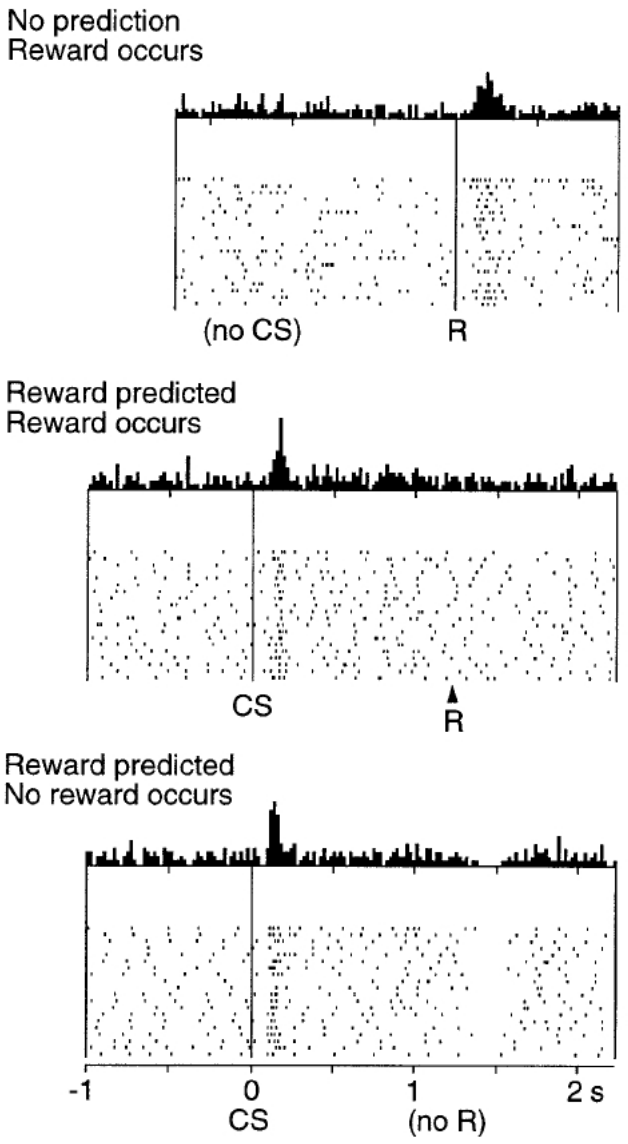
2.5 Neural Basis of Reinforcement Learning

Reinforcement Learning conceptually has its origins in Psychology, especially the model-free approach. Model-based on the other hand, owes its mathematical formulation mainly to Dynamic Programming and Optimal Control field. In this section we describe how a model-free approach managed to explain computationally the firing pattern of dopaminergic neurons. This was a very important attempt to connect the behavioral scale with the neural scale in the brain, and marked the RL framework as a formal theoretical tool for explaining various behavioral and neuronal patterns related to decision making.

The phasic activity of dopamine neurons has been found to resemble a type of learning algorithm signal (Montague et al., 1996; Schultz et al., 1997). These findings support the dopamine hypothesis, according to which neurons in the ventral tegmental area (VTA) and substantia nigra pars compacta (SNc) behave in accordance with reinforcement learning models, based on reward prediction error.

However, the dopamine neurons do not encode raw reward value directly. Instead, they encode the *difference* between an expectation of reward and reward received. Evidence of the firing pattern of dopamine neurons was provided by Schultz (1998), who examined the behavior of monkeys in classical conditioning tasks. This is illustrated in fig. 2.7.

One of the first attempts to computationally model this firing pattern was introduced by the simple conditioning model of Rescorla et al. (1972). This model



ht

Figure 2.7: Peri-stimulus time histogram (PSTH) of spikes from monkey neurons in VTA and SNc areas, while it was performing a simple conditioning task. **Top:** Prior to conditioning, the dopamine neurons fire immediately after an unexpected reward received. In a Rescorla-Wagner (RW) model ($\delta = r - \hat{r}$) there is no expectation of a reward and thus $\hat{r} = 0$ and $\delta = r > 0$ hence a positive error in the prediction of reward. **Middle:** After learning, the conditioned stimulus predicts reward, and the reward occurs according to the prediction hence there is no error in the prediction of reward $\hat{r} = r$ and $\delta = 0$. It seems that the dopamine neurons are activated by the presentation of the conditioned stimulus but fail to be activated by the predicted reward. **Bottom:** After learning, the conditioned stimulus predicts a reward but the rewards fails to occur and the dopamine neurons' activity appears to dip, reflecting the negative error in the prediction of reward ($r = 0$ and $\delta = -\hat{r} < 0$).

(R-W) is identical to the delta rule and simply states that the updating of weights w, which weight stimuli features, should be in the direction specified by the difference between actual r and estimated reward $\hat{r}$, δ

$$\Delta w = \delta \cdot x \tag{2.17}$$

and

$$\delta = r - \hat{r}, \text{ where } \hat{r} = \sum x \cdot w \tag{2.18}$$

where x is the input stimuli. The R-W model does not take future rewards into account, but only immediate rewards. To overcome this, a slightly modified model that is known as the Temporal Difference (TD) learning rule was established (Sutton and Barto, 1998; Sutton, 1988). The TD algorithm is well-suited algorithm to study the dopamine signal and its relation to synaptic plasticity, and consequently learning.

The TD algorithm is similar to the R-W model but it accounts for future rewards. Thus the modified delta rule, which specifies the direction in which the weights should change, is:

$$\delta = (r + f) - \hat{r} \tag{2.19}$$

where f represents the future reward that might be encountered. TD learning rule explains everything that R-W rule does, and more. For example, animals learn that a predictor of a predictor is also a predictor of reward which is second order conditioning. The R-W rule fails to capture second order conditioning as it accounts only for immediate rewards.

It is natural to wonder how a system can know what kinds of rewards are coming in the future. Experience of the past comes as the answer to this question. In order to predict future rewards a system takes account of the past knowledge, and uses it to make the best decision possible. Because knowledge of the sum of future rewards is not available, the TD method uses bootstrapped estimates of this sum, which evolve as a function of the difference between temporally successive predictions.

The TD method guides learning so that predicting future rewards become more accurate. It has been used successively to model various conditioning phenomena and its fit to the firing of dopamine neurons has led to significant research progress (for example see Daw et al. 2013). However not all aspects of dopamine cell activity can be modelled using traditional TD algorithms. Sometimes, rewards are not delivered immediately after an action is taken but they might be received with a

delay. Montague et al. (1996) introduced the tapped delay linear model to solve the problem. Daw et al. (2006) extended further the TD algorithm, by using Partially Observable Markov Decision Processes and semi-Markov dynamics, to incorporate variability in reward timing and allow for a greater range of state representations.

Despite the continuing efforts directed at establishing a precise role that dopamine might play in reward learning, a general theory that incorporates all aspects of dopamine's functions remains elusive.

2.6 Behavioral Evidence of Reinforcement Learning

To date, the RL framework has been used broadly to model the mechanism of how organisms select their actions in a constrained environmental context. In order to model a function, it is necessary that some simplifications and approximations be made. In this way, mathematical equations can explicitly describe decision making functions and then be evaluated. Computational modeling seeks to establish explanatory frameworks for cognitive or neural functions (Boden, 1988). It is based on the assumption that the brain processes information in an iterative 'algorithmic' way. In particular, cognitive functions are described as mathematical quantities that are 'learnt' in an iterative fashion, identical to that of a common computational algorithm. Usually such computations minimize or maximize quantities of external or internal importance (e.g., Dayan and Abbott 2005; Friston 2010). Such models also describe explicitly the variables of the process, and there are various methods that can fit such models to human (or animal) data in order to provide evidence for the appropriateness of the model and also for comparison with other similar models.

Apart from from neural evidence, from extracellular recordings in behaving animals and functional imaging of human decision making, of the existence of a key RL signal (temporal difference reward prediction error[3]) in the brain, there is evidence that humans and animals utilize a number of parallel decision making systems. Balleine et al. (2008; 1998) suggested that instrumental behavior is controlled by two learning mechanisms: a goal-directed one which consists of the acquisition of incentive value by the reward, and a stimulus-response habit mechanism that

[3]The reward prediction error δ, is defined as the difference between real outcome and predictive outcome from the previous trial: $\delta_t = outcome_t - prediction_{t-1}$.

involves learning about the instrumental contingency between the response and reward.

Daw et al. (2005) were one of the first to formalize the dual-action choice systems using the computational theory of reinforcement learning. Specifically, they suggested that the prefrontal circuit serves as a model-based reinforcement learning method, which is described by *dynamic programming* or *tree search* methods. In this method, a task is represented as a tree with all possible situations (states) that can be faced when engaged with the task. Each transition from state to state is realized by possible actions and their given rewards. In contrast, the dorsolateral striatum supports habitual or reflexive control and thus *reinforcement learning* is more appropriate to describe it. Such an approach represents only the expected future value Q for each action in each state, unpaired with future contingencies.

It is worth mentioning, that the models that were used in the study above were based on Bayesian RL (e.g., Dearden et al. 1999, 1998; Strens 2000). The full model was a dual-controller reinforcement learning model consisting of a model-free learner (Bayesian Q-learning) and a model-based learner (Bayesian tree search). With the Bayesian approach, Daw et al. (2005) tracked uncertainty about values of actions, along with the values themselves. The uncertainty exists because each system starts with little or no prior experience of the task. A controller[4] achieves dominance if the value provided by the controller has the least uncertainty. Finally, the probability of selecting an action is proportional to the "winning" value. It has to be noted, that for simplification the controllers are considered independent, contrary to other theories and architectures (e.g., Doya 1999; Sutton and Barto 1998), where the controllers interact (e.g., like the Dyna algorithm[5], Silver et al., 2008).

Daw et al. (2005) simulated the two-controller reinforcement learning model on a simple two-choice task, in which trained food-deprived rats performed a series of actions to obtain rewards. Their key quantitative results were consistent with qualitative expectations for the given task. By devaluing the reinforcer of an action, they observed that the model-free system continued to perform learned actions, consistent with the definition of habitual behavior (i.e., outcomes are not taken into account, except after long periods of training); in contrast, the model-based

[4] A function that controls action selection: in our case the model-based/model-free controllers.

[5] In this case though the problem was the balance between exploration and exploitation. As model free methods are used for experience sampling whereas the model based methods for planning, a hybrid system would attempt to optimally balance both.

system did not perform previously trained actions and it allowed for the prediction of the immediate outcome of an action or the long-term outcome of series of actions.

Other work (Daw et al., 2011) has further highlighted the interaction of these two systems in the two-stage task[6] common paradigm. Noting that both strategies were evident in their results, they suggested that action selection in their model is based on a weighted sum of both systems. Consistent with other findings from research on animal learning (Balleine and O'Doherty, 2010; Dickinson, 1985) they found evidence that the brain employs a combination of both strategies.

Both model-based and model-free systems have advantages and disadvantages. The model-based system appears more computationally costly in terms of time and neural resources, as it considers almost all contingencies from a given state. However, is more accurate and can adapt to environmental changes more rapidly. Thus, it should be used by an organism sparingly, such as whenever there are not enough 'data' from the model-free system. The model-free system is more efficient computationally, but requires more sampled experience from the environment. In cases where the organism has been under excessive training, or when few actions need to be evaluated, habitual responding is optimal in terms of time and resources. It is worth noting, that the model of the environment in model-based computations should either be given or learned (the agent uses an adaptive model of the reward and state transition dynamics in the form of an assumed probability distribution and updates it according to observations). Thus, it can adapt and change its decision tree quite quickly whereas the model-free system needs extensive training to achieve this.

Both systems feature statistical and computational properties that provide adaptive reasons for both to be employed within the same brain. The superior statistical efficiency of the model-based system might dominate the decision-making process in early instrumental trials. At this early stage, the complex noisy calculations of a model-free system, which are based on bootstrapping, are inefficient and a precise model-based mechanism would be favored. However, once the statistical inefficiency of bootstrapping has been overcome, a model-free response from the brain dominates the decision system.

The interaction of both systems is under active investigation. In particular, there is a general trend towards the hypothesis that model-based predictions train

[6]The common implementation of this, is to present 2 images which probabilistically lead to another two images which they lead to a scalar reward value. Usually the probabilities for each action-state transition are set to 0.7 and 0.3.

the model-free predictions either offline (e.g., during sleep Foster and Wilson 2007; Wilson and Foster 2006) or online (Doll et al., 2009; Gershman et al., 2014) or by providing prediction errors that can be used directly by the model free system (Daw et al., 2011).

Nature, from composite particles to living organisms, is governed by the principle of the least "energy". This means that every system tries to minimize energy consumption while satisfying other constraints (e.g., staying alive). This is also true for the neural system. In real life, humans and animals tend to create habits in order to 'save mental or physical resources'. On the other hand, when information/initial experience is not enough to create automated responses to specific stimuli, a planning system should be enabled for optimal action selection. For example, the more an athlete trains the more his responses become habitual and thus his actions are faster and more accurate. An experienced chess master could play almost habitually many games of chess, but in the face of a challenge he must evaluate all courses of action from his current state (although his decision tree would be much more pruned than an average player[7]). In sum, much of the progress in cognitive research to date is consistent with the existence of multiple decision-making systems in the brain.

2.7 General Discussion

Reinforcement Learning provides a formal framework for mathematically expressing decision making problems in the context of the maximization of a utility function. This utility function can represent time, points or abstract reward signals with binary (0 or 1) or continuous values. Therefore, all cases that involve optimal action selection (i.e., maximizing a reward function or achieving a goal) can be framed as MDPs and solved computationally.

As discussed in ch. 1, organisms (artificial or not) can be viewed as decision makers that attempt to select actions in order to maximize a specific reward function. The reward function can be positive/negative feedback from the environment that the agent interacts with, or an abstract reward that could account even for internal satisfaction/disatisfaction and progress. The RL framework in conjunction

[7]Experienced chess players have also demonstrated a better planning ability than average players (Unterrainer et al., 2006) in the Tower of London task (Shallice, 1982). This probably indicates that they are more skilled in searching decision trees for the best sequence of moves.

with this view enables us to analyze behavioral patterns observed when humans or animals are engaged in a various range of tasks. Such tasks include mental planning puzzles, video games, goal-driven tasks, navigation tasks, etc. Therefore, although seemingly mathematically abstract, RL can provide a tool that could extract useful insights on the computations involved in a decision making process.

Once the interaction of an agent with a task is converted into a MDP, specific solvers can be used to estimate optimal policies. The two main categories of solvers used throughout the thesis are Model-based and Model-free RL. Each one represents a different computation that takes place in the decision making system of the agent. In the first case, the agent has knowledge of the action-reward structures of the task, thus it can create a mental forward model - usually in the form of a decision tree - from its current state, accounting for all possible contingencies that it could encounter in the future. Then the computational problem would consist of selecting the best action from its current state given the decision tree. In the second case, the agent needs to learn how to take optimal decisions by gathering enough experience by a trial-and-error type of interaction with the task. The computation being performed in this case consists of iterations of exploratory or exploitative actions with the task, which ultimately would lead to an optimal behavior (policy).

In this chapter, we demonstrated how RL as a computational framework can create agents that generate optimal behaviors in various types of tasks. We also referred to how RL models could explain neural dopaminergic activity which links patterns from the neural scale to the behavioral scale. In the next chapter we will make use of this framework to describe mechanistically how actions are generated from parametrized RL) models, and how Bayesian model fitting methods may be used to determine values of these parameters in order to fit the models to human data.

Chapter 3

Reinforcement Learning II: Bayesian Fitting to Behavioral Data

ABSTRACT

Computational algorithms for reinforcement learning are based on a prediction error signal in order to approximate a reward function iteratively. Using such algorithms we can design computational models that represent quantitative hypotheses about how the brain approaches a problem and which are amenable to direct experimental testing. Trial-by-trial analyses of such data are suitable for developing a detailed and dynamic picture of learning. In this chapter we describe the computational approach used by researchers to analyze data from reward learning or decision making experiments, and illustrate it with examples. The whole process is described in detail by Daw (2011), but we summarize and illustrate it here as these methods are used extensively throughout the whole thesis.

3.1 Introduction

In a standard experiential decision experiment, such as a 'bandit' task, a subject is offered repeated opportunities to choose between multiple options (slot machines) and receives rewards according to her choice on each trial. The data usually consists of a series of choices and rewards from such a process. Computational theories claim that there is some kind of a relationship between the entire list of past choices and outcomes, and the next choice. Standard RL models (such as Q-Learning) assume restricted functions by which previous choices and feedback give rise to subsequent

choices. More specifically, the algorithm envisions that subjects track the expected reward from each slot machine, via some sort of running average over the feedback, and it is only through these aggregated "value" predictions that past feedback determines future choices.

It is also useful to separate a computational theory into two parts: the learning model, which describes the dynamics of the model's internal variables such as the reward expected from each slot machine, and the observation model which describes how the model's internal variables are reflected in observed data. Essentially, the latter model, regresses the learning model's internal variables onto the observed data (e.g., choices). In other words, it acts as the 'link function' in generalized linear modelling.

The observation models are typically noisy whereas the learning ones are deterministic. That is, given the internal variables produced by the learning model, an observation model assigns some probability to any possible observations. The "fit" of different learning models (or their parameters) to any observed data can be quantified statistically in terms of the probability they assign to the data. This procedure lies at the core of the methods that follow. In fig 3.1 we present a general 'recipe' that can be used in order to perform the same analysis presented in this chapter.

3.2 Choice Generative Processes

As we described in the previous chapter, we are interested in the mechanism of how a subject processes the available environmental information and makes decisions. A model describes this mechanism explicitly. This mechanism can be seen as a function f with a stimulus as the input and a choice as its output. Characterizing such a function as a mathematical object, for each of the N available subjects, we need to define a set of parameters θ_i where $i = 1 \dots N$. Eventually the set of choices $\mathbf{c}_i$ for each subject i, will be the output of a function of the model parameters: $c_i^j = f(\{\theta\}_i)$. This indicates that the jth choice of subject i was *generated* by the process f.

A *generative process* is a process where a set of outputs is realized by a parametrized function or a model. In fig. 3.2, we demonstrate with a graph such a process. The model, governed by its parameters θ_i, generates choices $\mathbf{c}_i$. At the top level, we assume that the model parameters are generated by another process which is

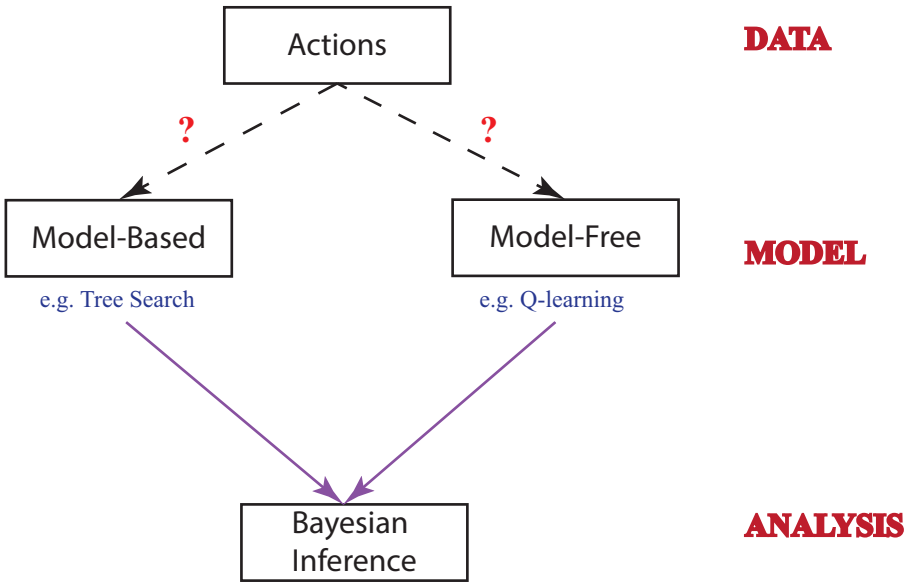


Figure 3.1: A general flowchart of the computational analysis used throughout the thesis. As discussed, Reinforcement Learning can be applied to model the behavior of a subject that is engaged with a task that is goal-driven (i.e., maximize the total reward). Once the actions of the subject have been collected we can start the process of behavioral analysis. First, a model will represent our hypothesis on the mechanism underlying the decision making process that generated his actions. If the model of the environment is known to the participant (i.e., all possible contingencies from every state of the task) then Model-Based methods should be preferred. In cases in which the participant learns by exploring/exploiting different choices without any explicit knowledge of the environment, then Model-Free methods should be used. Finally, under the model assumption we can perform Bayesian Inference to estimate the most likely parameters that generated the observed actions. Having the model parameters we can extract useful insights on the behavior of different population groups that participated in the experiment.

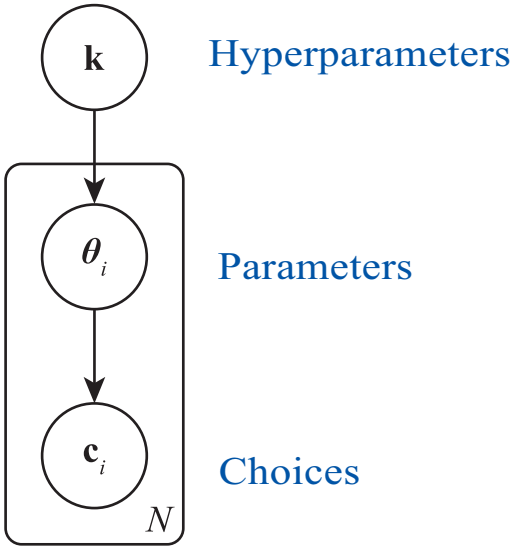


Figure 3.2: A graph representing a generative process. A function which is parametrized by a set of θ_i parameters, represents the mechanism we attempt to model, receives as input a set of stimuli and generates choices. If the model parameters are assumed to be generated from another stochastic process, parametrized by $\mathbf{k}$, then we form an hierarchical model and the parameters at the top level are usually referred to as *hyperparameters*.

parametrized by hyperparameters $\mathbf{k}$.

The process of *inference* follows exactly the opposite direction of a generative process: from bottom to top. This means that having the observed choices made by a subject, we need to hypothesize one or more forms of a mechanism that presumably generated the choices. Parameter estimation consists of inferring the parameters of this hypothesized mechanism and then comparing multiple forms of it, eventually keeping the one that describes the data the best.

In the following sections we will demonstrate how such an inferential process is realized, by using three Bayesian approaches: Maximum Likelihood (ML), Maximum a Posteriori (MAP) and Markov Chain Monte Carlo estimation (MCMC).

3.3 Parameter Estimation by Maximum Likelihood

Model parameters can characterize a variety of scientifically interesting quantities such as how quickly subjects learn and how sensitive they are to different rewards

or punishments. Here we discuss how we can estimate the parameters of a RL model $\mathcal{M}$ in a Bayesian framework using data (action sequences) collected from a simple example. This will enable us to extract behavioral insights by linking the observed action sequences to the model parameters. Therefore, we satisfy our initial assumption that the actions were generated by a model with specific parameters and that model is responsible for the observed behavior. All methods, the simulation example and the inference scheme were coded in MATLAB by the author of this thesis exclusively.

Suppose that we have some free parameters $\boldsymbol{\theta}_{\mathcal{M}}$ that parametrize our model. This model (the composite of our learning and observation models) describes a probability distribution over possible data sets $\mathcal{D}$, or likelihood function $P(\mathcal{D}|\mathcal{M}, \boldsymbol{\theta}_{\mathcal{M}})$. According to Bayes rule

$$P(\boldsymbol{\theta}_{\mathcal{M}}|\mathcal{D}, \mathcal{M}) \propto P(\mathcal{D}|\mathcal{M}, \boldsymbol{\theta}_{\mathcal{M}})\, P(\boldsymbol{\theta}_{\mathcal{M}}|\mathcal{M}) \tag{3.1}$$

the posterior probability of the free parameters, given some data, is given by the product of the likelihood of the data, given the parameters (our assumptions about how the model describes the processes that we want to model) with a prior probability over the parameters. In a maximum likelihood framework (as also in the maximum a posterior approach (MAP) that we describe below) we seek a point estimate of the parameters rather than a distribution over all possible parameter values. This setting of the parameters is the one that maximizes the likelihood function and is the one that is most likely to be used by the model $\mathcal{M}$ to generate the data. This optimum set of parameters is denoted by $\hat{\boldsymbol{\theta}}_{\mathcal{M}}$.

We illustrate the above with a simple example. Consider a two-arm bandit problem, where we need to choose sequentially between two slots machines ($c_t \in \{L, R\}$) in order to obtain maximum accumulated reward after some trials T, $t = 1 : T$. In this example a reward $r_t \in \{0, 1\}$ is received stochastically at each trial. According to a simple Q-learning model, the subject assigns an expected value to each machine $(Q_t(L), Q_t(R))$ at each trial. These values are initialized with 0 and the value for the chosen slot machine is updated according to the learning rule

$$Q_{t+1} = Q_t(c_t) + \alpha \cdot \delta_t \tag{3.2}$$

where $0 \leq \alpha \leq 1$is a free learning rate parameter, and $\delta_t = r_t - Q_t(c_t)$ is the prediction error. Next, we assume an observation model according to which the subject

makes its choices c_t depending on the Q_t values. One of the most common approaches to subject's action selection is the softmax distribution

$$P(c_t = L|Q_t(L), Q_t(R)) = \frac{\exp(\beta \cdot Q_t(L))}{\exp(\beta \cdot Q_t(R)) + \exp(\beta \cdot Q_t(L))} \tag{3.3}$$

It is worth noting, that this equation defines a policy, which is a mapping from states to actions, as defined in Chapter 1.

The parameter β is known as the inverse temperature parameter or reward sensitivity, and its role is to tune the amount of exploration/exploitation in subjects' choices. Eq. 3.3 is also equivalent to logistic regression (a type of generalized linear model, GLM), in which the dependent variable is the binary choice c_t and the predictor variable is the difference in values $Q_t(L) - Q_t(R)$ with β as the regression weight, connecting the Qs to the choices. For more than 2 choices the logistic regression generalizes to the multinomial logistic regression.

Although we noted the equivalence of eq. 3.3 with logistic regression, it is not possible to use the well-studied methods for regression of generalized linear models. This is because although the observation stage of the model represents a logistic regression from values Q_t to choices c_t, the values are not fixed but are dependent on the parameter α of the learning process. As this does not enter the likelihood linearly, they cannot be estimated by a generalized linear model, and we must search for the full set of parameters that optimize the likelihood function. It is straightforward to write a function that takes in a dataset (a sequence of choices $c_{1:T}$ and rewards $r_{1:T}$) and a candidate setting of the free parameters, loops over the data computing equations 3.2 and 3.3, and returns the aggregate likelihood of the data. In our experiment the likelihood is just the product of the probability of each choice at each trial, given by

$$\mathcal{L}(\alpha, \beta) = \prod_{t=1}^{T} P(c_t|Q_t(L), Q_t(R)) \tag{3.4}$$

It is worth mentioning that the quantity in eq. 3.4 is often an exceptionally small number and it is numerically more stable to compute its log. This computational problem needs to be taken into consideration during algorithmic implementations as it might lead to extreme estimations of the parameters. Since the logarithmic function is monotonic, this quantity has the same optimum as the non-logarithmic case, but it is less likely to underflow the minimum floating point value of a com-

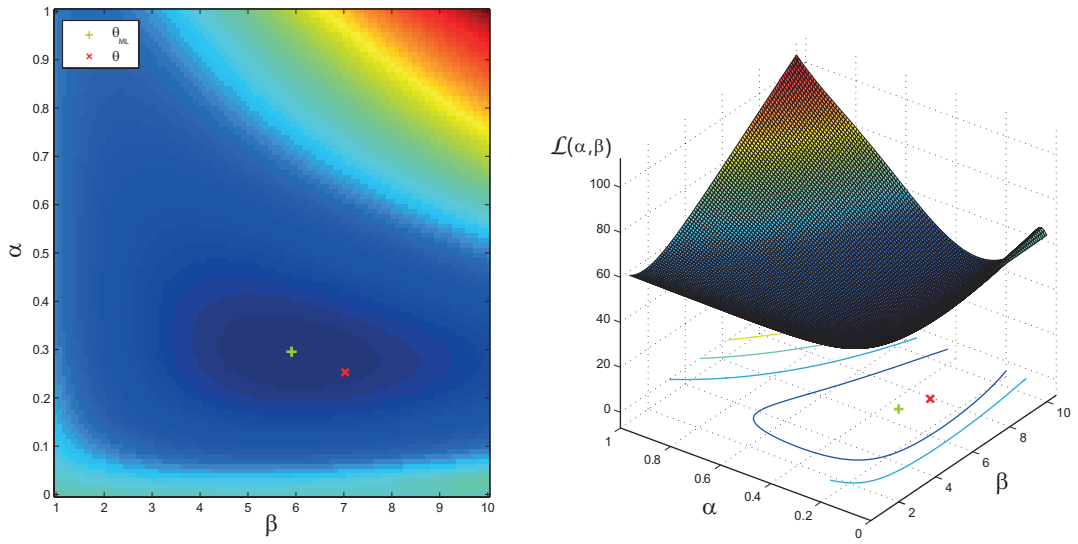


Figure 3.3: Parameter space search method

puter. In general, optimizers in various software packages tend to minimize a function, and thus, in our parameter estimation procedures, we attempt to minimize the negative log likelihood of eq. 3.4.

In order to compute the likelihood, one tempting approach is to discretize the parameter space and compute the likelihood everywhere and simply search for the best one. This approach is illustrated in fig. 3.3.

Note that this is the negative log-likelihood function estimated at a specified range of the parameters. The data were generated from a simulation of 100 trials with parameters $\alpha = 0.25$ and $\beta = 7$. The estimated set of parameters, computed at the minimum of the negative log-likelihood, was $\hat{\alpha} = 0.30$ and $\hat{\beta} = 5.91$. A graphical analysis of the whole procedure is given in fig. 3.4. We can see in the analysis of the probability of each choice, how the probability of each action changes, as the algorithm learns which action is more rewarding.

Each row of fig. 3.4 represents the characteristics of a specific choice (left or right slot machine in our case). In the left column, the plots represent the probabilities of changes during the trials. If the participant chose the corresponding alternative then at the specific trial a vertical dot line is plotted. In the right column the value function, according to each trial, is plotted in the same diagram with the reward given for the particular action during each trial. The plots illustrate how the subject selects an action according to the forthcoming reward. The value functions indicate to the subject that the left or right machine is most likely to provide a

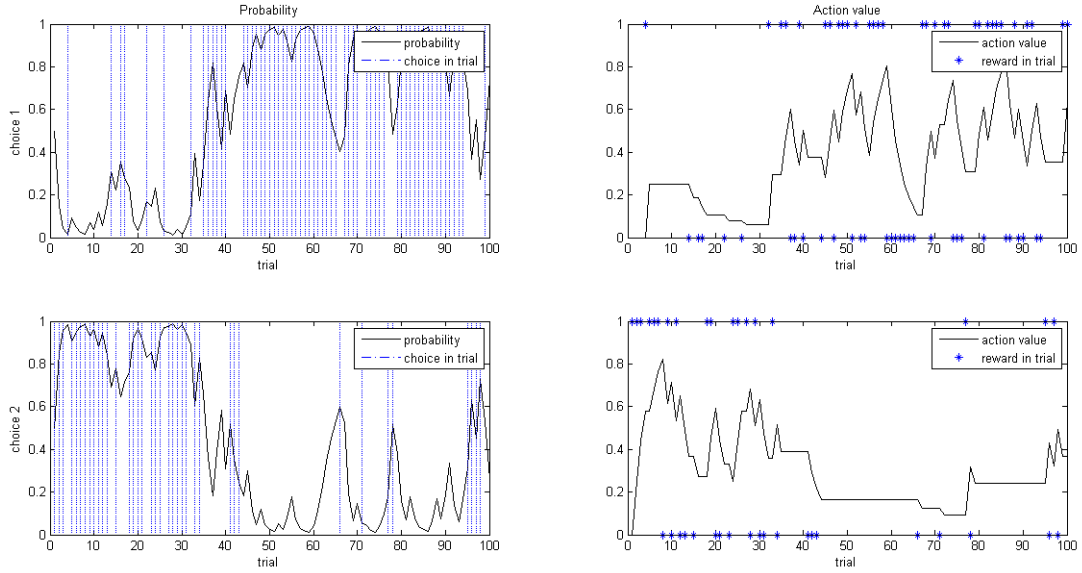


Figure 3.4: Analysis of action selection procedure.

reward at each trial. Confidence intervals of the parameters can be estimated from the inverse Hessian of the negative log-likelihood computed at the optimum $\hat{\theta}_{\mathcal{M}}$ for further statistical analysis of the model.

Although the above approach seems tempting, it often leads to poor results for many reasons (coupled parameters, inappropriate ranges, etc.). In addition, the models that are typically used have many parameters. To avoid such errors nonlinear function optimization is usually preferred. MATLAB functions such as `fmincon` or `fminsearch` may be used to find the single best setting of the parameters. This process, however, is not as automatic as it sounds. The parameter estimation via function approximation needs supplementary information such as the gradient and Hessian of the likelihood, parameter boundaries and tuning.

For the above example the nonlinear function approximation gave $\hat{\alpha} = 0.29$ and $\hat{\beta} = 5.94$. In fig. 3.5 we can see the path from the initial conditions ($\alpha = 0$, $\beta = 1$) until the minimum $(\hat{\alpha}, \hat{\beta})$. The likelihood might not have a global maxima and the whole procedure is dependent on the initial conditions, thus it is advisable to try different sets of initial conditions and select the ones that optimize the likelihood.

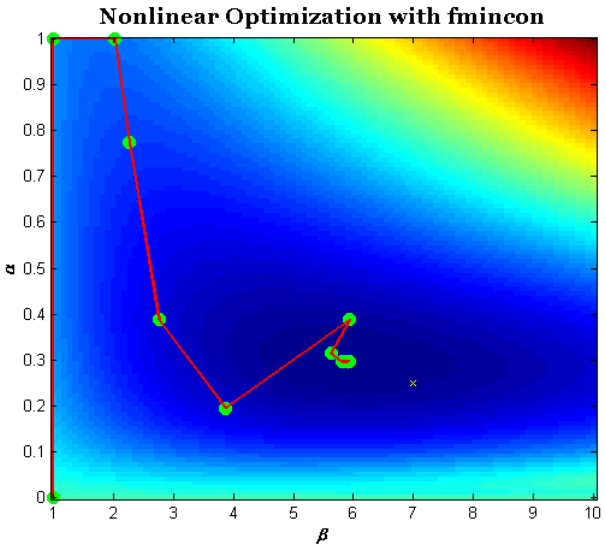


Figure 3.5: Gradient path to the solution. The heat map represents the likelihood $\mathcal{L}(\alpha, \beta)$ surface and the two axis the corresponding parameters of it. The cross symbol in th egraph denotes the true parameter values that were used to generate the data.

3.4 Parameter Estimation by Maximum a Posteriori

For a fully Bayesian treatment it is reasonable to consider a suitable prior $P(\boldsymbol{\theta}_{\mathcal{M}}|\mathcal{M})$ for the parameters. The prior can also be a hard or smooth constraint on the range of the parameters (uniform or Gaussian distribution) and regularize their estimates at the optimization procedure, now called maximum a posteriori (MAP). More informative priors can be selected from population models after processing data from particular subjects' groups. In this case hierarchical models of the parameters can be considered and inference of these can be implemented again with various Bayesian approaches. Problems might arise during the optimization of the posterior, and sampling methods such as Markov Chains Monte Carlo (MCMC) can be useful alternatives of the MATLAB function `fmincon`.

Here we assume that the two parameters are drawn from two Gaussians

$$\alpha \sim \mathcal{N}(\mu_\alpha, \sigma_\alpha) \tag{3.5}$$

$$\beta \sim \mathcal{N}(\mu_\beta, \sigma_\beta) \tag{3.6}$$

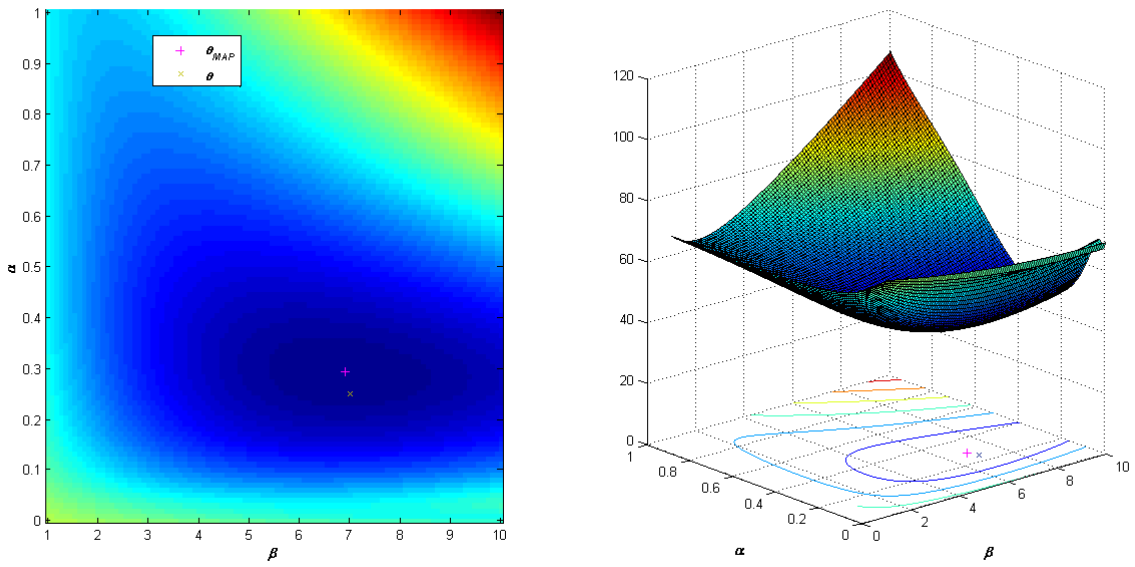


Figure 3.6: Maximum a posteriori estimation by parameter space search.

with $\mu_\alpha = 0.5$, $\mu_\beta = 10$, and standard deviations $\sigma_\alpha = 1/30$ and $\sigma_\beta = 1/5$.

We have to mention that this whole process is for demonstration of the method and not to extract insightful conclusions about subjects' behavior, as the data were simulated artificially. However, the choice of the parameters of the Gaussians has to do with the range of the parameters, and after some tuning the results are better than the maximum likelihood method, as expected.

The hyperparameters (the parameters of the priors) can be inferred by Bayesian population statistics, though here we chose them by hand. As we did in the case of maximum likelihood, we minimize the negative log-posterior with MATLAB's nonlinear optimizer. We also search for optimum solutions using a simple search of parameter space.

First we search the parameter space as we did before (fig. 3.6). As we can observe, the MAP estimation ($\hat{\alpha} = 0.29$ and $\hat{\beta} = 6.91$) is much closer to the real values of the parameters, for the reason that the priors provide extra information about them. The second step is to optimize the posterior with the function `fmincon` of MATLAB. The procedure gave $\hat{\alpha} = 0.28$ and $\hat{\beta} = 6.89$. Finally, in the next section we examine MCMC method for sampling from the posterior.

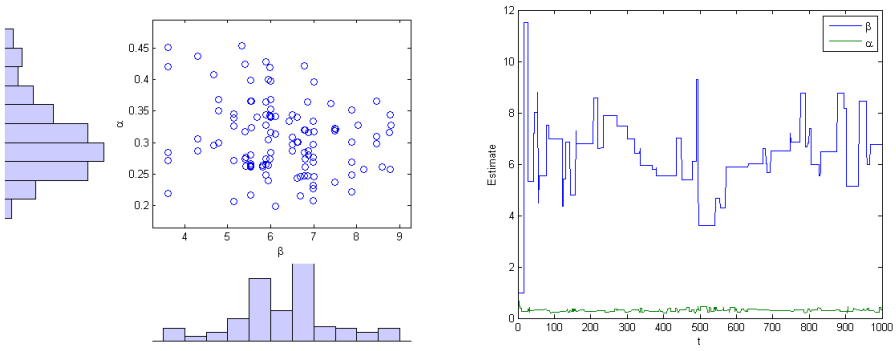


Figure 3.7: MCMC sampling using the Metropolis-Hastings algorithm.

3.5 Markov Chain Monte Carlo (MCMC) Estimation

The Markov Chain Monte Carlo sampling methods are very popular as inference methods. One of the reasons that we would need sampling methods is that the posterior distribution over parameters (assuming that the parameters are independent)

$$P(\alpha,\beta|\mathcal{D},\mathcal{M}) \propto P(\mathcal{D}|\mathcal{M},\alpha,\beta)P(\alpha)P(\beta) = \\ = \left[\prod_{t=1}^{T} P(c_t|Q_t(L),Q_t(R))\right] \cdot \mathcal{N}(\mu_\alpha,\sigma_\alpha) \cdot \mathcal{N}(\mu_\beta,\sigma_\beta) \quad (3.7)$$

is not amenable to analytic techniques. For this reason we draw samples from this posterior distribution.

Instead of trying to optimize the a posteriori function, we sample from it. If we have enough samples, the distribution of them will resemble the posterior distribution over our parameters and thus we can determine the mode of it and so the optimum parameters. Here we demonstrate a simple MCMC algorithm, the component-wise version (i.e., where we sample each dimension separately) of the Metropolis-Hastings (MH) (Hastings, 1970; Metropolis et al., 1953) algorithm (alg. 3.1).

The estimated parameter values were $\hat{\alpha} = 0.30$ and $\hat{\beta} = 6.32$ and the corresponding procedure is shown in fig. 3.7. As we can, see the proposed model does not describe the process sufficiently. This process needs a lot of tuning. For example, the proposed distributions that we sample should have appropriately chosen pa-

Algorithm 3.1 Metropolis-Hastings (MH) Algorithm

Set t=1
Initialise α and β
repeat
  $t = t+1$
  Do a MH step on α:
  Generate a proposal $\alpha*$ from a suitable distribution (we choose Gamma)
  Evaluate the acceptance probability using α^*, α, and β
  Generate a u from a $Uniform(0,1)$ distribution
  if $u \leq$ acceptance **then**
    accept the proposal and set $\alpha = \alpha^*$
  end if

  Do a MH step on β:
  Generate a proposal $\beta*$ from a suitable distribution (we choose Gamma)
  Evaluate the acceptance probability using β^*, β, and α
  Generate a u from a $Uniform(0,1)$ distribution
  if $u \leq$ acceptance **then**
    accept the proposal and set $\beta = \beta^*$
  end if
until $t = T$

rameters. However, as we mentioned, the whole procedure was for demonstration purposes only.

3.6 General Discussion

The approach described above follows a Machine Learning perspective, in which a learning process generated a data set of observations $\mathcal{D}$ and we attempt to find a mathematical description of it. Specifically, we assumed that our data (actions) were generated by a model, the learning rule in eq. 3.2 used by the action selection equation 3.3 which generates actions. Apparently, the inference problem (i.e., finding the parameters of the model that most likely generated the data) is reduced to a logistic regression problem. Thus, the β parameter has the role of the regression coefficient in a logistic regression setting. In this sense, the Q values for every action in a given state will be the 'data' in the logistic regression case. However, the 'data' are not constant values, as in the case of the logistic regression, and change across time according to the learning rule (eq. 3.2). Moreover, they are dependent on the learning parameter α.

Obviously, in a logistic regression setting we are interested in inferring the regression coefficients which in our case will be represented by the parameter β. As we discussed, our 'data', which are reflected in the Q values, are dependent on another parameter, the learning rate α. This coupling of the two parameters creates problems in the inference. For example, a poor search on one will also corrupt the estimates for other parameters.

Point estimates of parameters, such as ML and MAP suffer from the common problems of such approaches. With ML estimation we seek the parameter θ that maximizes the probability of the data given that parameter (i.e., maximizing $P(\mathcal{D}|\theta)$). This approach, however takes no account of any information that we might have about the range of this parameter and eventually will lead to unregularized solutions. A prior information on the parameters' constraints, indicating their likely range, is incorporated with MAP estimation and leads to regularized solutions. With this method we seek the parameter that maximizes the posterior probability ($P(\theta|\mathcal{D})$) over possible parameters in the light of the observed data.

With these methods, we characterize the posterior distribution by its mode and, although this seems very useful, it has some drawbacks. For example, there might be other set of parameters that lead to high values of the posterior distribution. It is reasonable to seek these parameters too. Moreover, there might be a correlation between parameters which may occur in multimodal posterior distributions. With ML or MAP we estimate a single set of parameter values for a model, and thus these methods lead to point estimates that characterize a probability distribution rather than the probability distribution itself. To achieve this we need posterior sampling.

The reason for using MCMC to estimate parameters is to characterize the full posterior distribution and not only its mode. In some cases, we might be able to derive analytical expressions of the posterior distribution but usually this is not the case, and we have to resort to sampling techniques such as MCMC. Having a characterization of the posterior distribution we are able to calculate means, variances and other moments of the distribution.

Ultimately, model selection methods can be used in order to evaluate which model fits the data best (i.e., the best action selection method that describes subject's decision process), and which distributions to use for priors. For example, in the RL framework, we can test hypotheses such as whether the subjects learn according to eq. 3.3 or whether they evaluate actions by learning more fundamental

facts about the task and reasoning about them (model-based RL).

The aforementioned processes can be applied in other data sets as well (fMRI, neural spikes, etc.). Furthermore the learning rule and action selection rule can be changed according to the specific problem. Another approach that is commonly used is the Kalman filter. In this approach, the fitting is done online and the learning rate can change during each trial.

It is beyond the scope of this thesis to compare different Bayesian methods for estimating posterior distributions and readers are referred to Bishop (2006) or Gelman et al. (2014) for an in-depth and detailed treatment on these matters. Our intentions were to illustrate the link between the machine learning techniques aiming to extract patterns from data, and how these can be applied in combination with learning and decision making mechanisms.

To conclude, parameter inference is feasible but not trivial or automatic, and the whole process needs proper tuning and monitoring. It is common for the inference program to return odd parameter estimates due to issues such as numerical stability, problematic reward values and parameter boundaries. Apart from the methods discussed here, it is necessary for the experimenter to have a substantial amount of data to test such models.

In the following chapter we put all of these methods into practice by examining various aspects of cognitive search or planning. Planning can be represented in a form of a decision tree with all possible contingencies represented as the tree nodes and branches. We will formulate such types of decision problems as MDPs and attempt to solve them with RL. Furthermore, we will infer model parameters and examine the cognitive links of these parameters using the techniques described in this chapter, and consider how they can be related to developmental science.

Chapter 4

Model-based Analysis of Mental Planning

ABSTRACT

As discussed in previous chapters, one of the two main decision making systems thought to be responsible for goal directed behavior and is linked to model-based RL approaches. This section will first briefly review some evidence of goal-directed decision making in young children. Thereafter, we present in detail how computational model-based RL models can account for the behavioral phenomena observed in mental planning tasks. The main question we ask is related to how the planning process functions in tasks in which the reward is given only when the task is solved, and there is no step-by-step guidance during the solution. Furthermore, we are interested in explaining differences in the way humans' cognitive search functions given their age. To relate the above questions to behavioral data, we employed the Tower of London task and a task that demands navigating among different states in order to maximize the collection of a form of reward. We introduce a shaping reward scheme for model-based RL in the case of sparse rewards and report an analysis on the way humans at different ages prune their decision trees.

4.1 Introduction

In the previous chapters we discussed evidence of two distinct learning systems in the human and animal brain: Stimulus-Response (S-R) learning, which is linked to model-free RL approaches, and Response-Outcome (R-O) learning, which is linked to model-based RL methods. In S-R learning, the behavior learned is considered habitual - for example, a rat might associate the press of a lever with a rewarding outcome, and perceive pressing the lever as a desirable action. In R-O learning, the

behavior is characterized as goal-directed - in our example, the rat associates the pressing of a lever with a positive outcome. The goal is to obtain the reward, thus as long as it desires that outcome it will keep pressing the lever. The emphasis — or, in other words, the learning cue — in the first example is on the action that was associated with a reward, and even with the absence of the reward the trained rat will keep pressing the lever. In contrast, in the second case the learning cue is the outcome, as the lever press is associated with a positive outcome and thus in the devaluation or absence of the positive outcome, the rat will stop pressing the lever. The main focus of this chapter is the second case and how this is realized computationally in the decision making process of humans.

In the domain of human and animal learning, outcome revaluation (which refers to changing either state transitions or the value of the rewarding outcome) has been employed to determine whether selection of a response is affected by its outcome. To determine the role of R-O learning in the control of young children's instrumental behavior, Klossek et al. (2008) trained 18- to 48-month-old children to manipulate visual icons on a touch-sensitive display to observe different types of video clips as outcomes. After the training phase, one of the outcomes was devalued by repeated exposure, and children's propensity to perform the trained actions was tested in an extinction paradigm (i.e., until the disappearance of the learned behavior because of the lack of reinforcement of this behavior).

At test, children older than 2.5 years performed the action trained with the devalued outcome less than other children who were trained with a valued outcome, thereby demonstrating that their actions were mediated by action-outcome learning. This means that the older children were able to adjust their preferences relevant to the outcome, which is sign of goal-directed behavior. By contrast, the responses of younger children (mean age < 2 years) were resistant to outcome devaluation and may have been elicited directly by the icons associated with each response, rather than mediated by a specific response-outcome expectation. As discussed in Chapter 2, model-free RL involves prediction of future rewards based on a value function that reflects accumulated past experience, without encoding the identity of an outcome. Therefore, behavior that is governed by such a system is unable to adapt immediately to changes in the outcome.

Klossek et al. (2011) further investigated the importance of choice training in enhancing the sensitivity to outcome value, and established its critical role in maintaining goal-directed behavior, not only in rats but also in humans. The

instrumental response that is sensitive to its outcome value is controlled by a tree-search mechanism that generates value predictions. The tree-search mechanism implements a forward model by simulating future states that are associated with various available response alternatives. If circumstances change, this mechanism can adapt immediately as it is based on instant-by-instant predictions of a specific outcome value. The authors concluded that their experimental results suggest that training an agent in a free choice between two actions which yield different outcomes benefits goal-directed action control.

Similar results were acquired by Kenward et al. (2009), in which 24-month-old toddlers learned to retrieve an object from a box by pressing a button, and then the object's value was increased by allowing the child to play a game with the toy obtained from the box in which the specific toy was necessary. After the object's subsequent disappearance, these toddlers attempted more frequently to press the button to in order to retrieve the object as compared to another group of 24-month-olds who had learned to retrieve the object but for whom the object's value was unchanged (i.e., at the play phase the toy was different from the one obtained from the box, meaning that the value of the object in the box remained at baseline). They tested whether the the first group's tendency to press the button in order to retrieve the object from the box, was increased as compared to the second group, as the object became more desirable. Their experimental results were consistent with Klossek et al. (2008; 2011), showing that the sensitivity to outcome value when selecting actions influences decision-making in young toddlers.

The studies of goal-directed behavior described so far have a common characteristic: a one-step look-ahead mechanism that relates a particular response to a particular outcome. However, optimal sequential decision making requires a mechanism that can look ahead many steps into the possible contingencies, and apparently would be much more complex. Mental planning (or cognitive search) is an executive function that does exactly this and is central to goal-directed behaviour, in any task that requires the organization of a series of actions aimed at achieving a goal.

In this Chapter we use the Tower of London task (described in section 4.3) as a domain for testing and modeling the planing mechanism employed by humans. Although the planning mechanism used in this task has been investigated thoroughly (e.g. Albert and Steinberg, 2011; Baughman and Cooper, 2007; Bull et al., 2004; Newman et al., 2003; Phillips et al., 2001; Shallice, 1982), to our knowledge,

there are few links to computational modeling within the RL framework. We employ three reinforcement learning models of planning, and use Bayesian analysis to fit each model to data from humans performing the ToL task. The data were obtained from previous studies of children and adults tested in the ToL task and in a computerized version of it, respectively. The same analysis was conducted using data from children tested in a spatial navigation task (the Planet task), in which participants had to choose optimal paths between destinations to maximize their total reward. Our aim is to investigate further the mechanisms underlying planning processes, and to contribute theoretically to RL models regarding these processes.

4.2 Methods

4.2.1 Model-based analysis

Consider a task in which a rat at time-step t is in state s_t facing two different types of levers, and has to decide which one to press (choices, c_1 and c_2). The first choice c_1 leads to a new state s_1 or s_2 with probability 0.3 and 0.7 respectively. At state s_2 the rat receives a piece of cheese ($r = 1$). The second choice c_2 leads to states s_3 and s_4 with probability 0.3 and 0.7 respectively, and to no reward. Fig. 4.1 shows a schematic representation of this task. One way for the rat to decide which lever to press, is to evaluate how good each future state is by exhaustively searching all possible choice outcomes it has, from the given state.

The above choice evaluation approach involves traversing a sort of associative chain, determining that the lever press is worthwhile via its association with cheese. This approach represents a very simple model-based approach. The environmental model is considered known (i.e., the state transitions and rewards) after some training experience. With this information, the rat can simulate possible action-outcome contingencies and decide accordingly what to do when it is in state s_t. In the event of reward devaluation, or any general change in the environmental model, the rat can use its forward model to take these changes into account and again act optimally (i.e., earn more cheese).

Models of S-R habitual behavior cannot capture changes in the environment without extensive retraining under the new conditions. The reason for this is that the changes cannot be reflected back to the rat's decision system. A model-

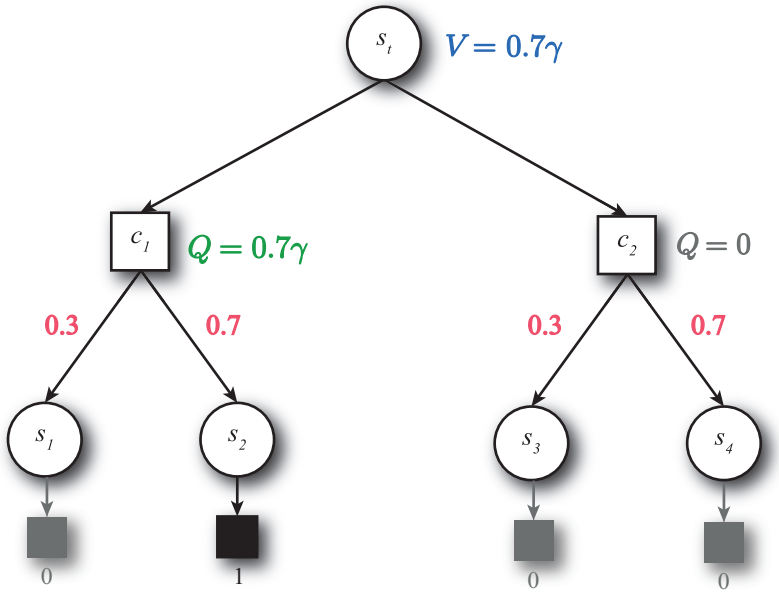


Figure 4.1: Model Based Analysis. A schematic representation of model-based learning. Here the model is known (rewards and transitions), thus from state s_t all possible outcomes can be evaluated. The circular nodes represent states and the square ones action-state nodes. Starting from the bottom of the tree and going up, the first action-state node will give $Q(s_t, c_1) = \gamma \sum_{s' \in \{s_1, s_2\}} \mathcal{P}(s'|s_t, \alpha_1)\mathcal{R}(s_t, \alpha_1) = 0.3 \cdot 0 \cdot \gamma + 0.7 \cdot \gamma = 0.7\gamma$. Similarly, we can compute the second action-state node. At the top state node we compute $V = \max\{0.7\gamma, 0\gamma\} = 0.7\gamma$. Calculations are based on equations 2.9 and 2.11.

based agent, however, chooses actions using an internal model of the environment: this includes which actions in which states lead to certain outcomes and a reward function that represents the outcomes of this chain of action and states. These choices will adjust automatically in response to changes of the dynamics of the environment, such as state-transition changes and reward devaluations.

In the rat paradigm, the rat maintains an internal representation of the environment (i.e., environmental model which consists of transition probabilities and a reward distribution) which it can use to reflect back any changes it observes. From this point it can plan anew its actions, by simulating possible action trajectories based on this internal representation of the environment. A tree structure can represent all the possible courses of actions that the rat can take in a computationally convenient way.

For the reasons we described, model-based approaches are linked to goal-directed behavior and are appropriate to model cognitive search and animal planning pro-

cesses. We need to stress that the above simple example is limited and is described here in order to give the reader an illustration of model-based approaches. There are other interesting cases where the MDP (sec. 2.2.1) is unknown to the agent and the problem of balancing exploration and exploitation (sec. 2.3) is much more apparent. Furthermore, the above example can be easily solved by dynamic programming methods such as value iteration methods. However, when the task consists of a large number of states, then evaluating the whole decision tree, even with a given horizon, is not a computationally efficient solution. To address this problem, in this chapter, we present various pruning models which aim to decrease the size of the decision tree.

In the next section, we describe three model-based RL models used to describe the mechanism underlying planning (fig. 4.2). The models, the model fitting and the model comparison procedures are described in detail in Huys et al. (2012) but we repeat the description here for completeness. All three models assume that subjects choose actions stochastically, with the probability of choosing action (or choice) c_t from state s_t at time t given by:

$$p(c_t|s_t) = \frac{e^{\beta Q(s_t,c_t)}}{\sum_{c'} e^{\beta Q(s_t,c')}} \tag{4.1}$$

The parameter β is an inverse temperature that represents the agent's sensitivity to rewards. The higher the value of β, the more probable it is for the agent to choose the action that maximizes the current Q function. Otherwise the agent will choose a non-optimal action (according to its current evaluation of the decision tree). Naturally, this leads to exploration of other actions and therefore the parameter β can be seen as mediator of exploration and exploitation. The three models that will be used throughout the chapter differ in the calculation of the function $Q(s_t, c_t)$.

The analysis that we will use here, as we mentioned, is model-based RL. This means that the agent pursues a goal and thus that it is following a R-O learning scheme. The first model is the *Lookahead* model. This model is simply a tree search model in the sense that it searches all available options until the end of the tree:

$$Q_{lo}(s,c) = R(s,c) + \max_{c'} Q_{lo}(s',c') \tag{4.2}$$

where s' is the successor state from state s after selecting choice c.

For a problem with a large action and state space, the evaluation of the whole

decision tree is computationally costly. Furthermore, we assume that humans have a limit to the depth and breadth of their simulated decision trees. Thus, there must be a pruning process that removes certain possible trajectories in the tree, in order for the decision process to be efficient. We explore this process in the second model, the *Discount* model.

In the Discount model, we assume that at each level[1] of the decision tree, a biased coin is flipped in order to determine whether the tree search should be terminated and return zero reward or proceed to the next level. Let the probability of stopping be γ, the Q values are estimated by:

$$Q_d(s,c) = R(s,c) + (1-\gamma)\max_{c'} Q_d(s',c') \tag{4.3}$$

Then the future outcome, k steps ahead, is weighted by the probability $(1-\gamma)^{k-1}$ that it is encountered.

The third model we implemented is a modification of the Discount model, which we refer to as the *Pruning* model. Originally, this model was used to stochastically stop the tree search at a point where a big penalty was perceived. For reasons that will be apparent later in this chapter, we will assume that an agent that tries to achieve a goal state, will tend to avoid states with large dissimilarity with that goal state. Thus, we modify the calculation of Q from the Discount model to the following:

$$Q_{pr}(s,c) = R(s,c) + (1-x)\max_{c'} Q_{pr}(s',c') \tag{4.4}$$

where

$$x = \begin{cases} \gamma_S & \text{if } R(s,c)\text{is a large negative reinforcement} \\ \gamma_G & \text{else} \end{cases} \tag{4.5}$$

γ_S (Specific pruning parameter) is the probability that the agent stops evaluating the decision tree while it is at a state in which the immediate reward leads to a subsequent state with great dissimilarity with the goal state. γ_G (General pruning parameter) is γ as in the Discount model.

[1]The first level is the root state and the immediate successor states. The second level consists of these states and their immediate successor states and so on.

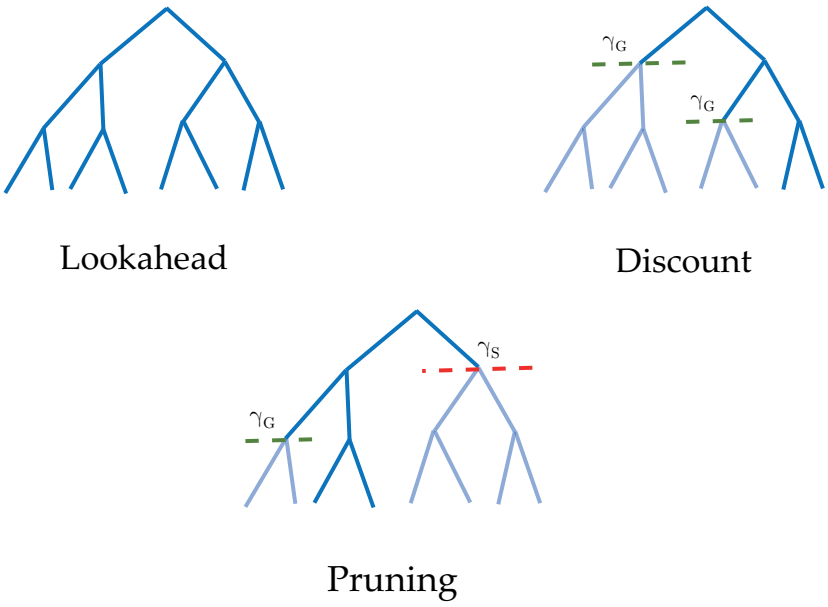


Figure 4.2: The three planning models used for the analysis. The planning process starts at a (real world) state, the root state, and forwardly simulates all possible trajectories from the root state, given a reward and transition function. **Top left**: the *Lookahead* model, evaluates the whole decision tree following the root state. **Top right**: the *Discount* model, probabilitstically terminates the tree expansion on every visited state. B**ottom**: the *Pruning* model, combines the pruning procedure from the Discount model with an extra stochastic pruning parameter, which terminates the tree expansion from a state where great dissimilarity with the goal state is encountered.

4.2.2 Model fitting procedure

Because we are interested in developmental applications of the above models, we will test them in populations of different age. For this purpose, assume a hierarchical model that describes how data are generated for each age group (fig. 4.3). As described by Huys et al. (2012), each model is characterized by a set of parameters $\mathbf{k}_i$, for each subject i, that are generated by a Gaussian distribution $\mathbf{k}_i \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{v}^2)$ with parameters $\boldsymbol{\theta} = \left\{\boldsymbol{\mu}, \mathbf{v}^2\right\}$. We will refer to these as *hyperparameters*. The whole analysis is applied separately for each of the age groups. We fit the model parameters and the hyperparameters in a joint scheme, using the EM algorithm (Dempster et al., 1977), maximizing the marginal likelihood given all data by all N subjects:

$$\hat{\boldsymbol{\theta}}^{ML} = \arg\max_{\boldsymbol{\theta}} P(\mathcal{C}|\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \left(\prod_i^N \int P(\mathbf{c}_i|\mathbf{k}_i) P(\mathbf{k}_i|\boldsymbol{\theta}) d^N \mathbf{k}_i \right) \tag{4.6}$$

where $\mathcal{C} = \{\mathbf{c}_i\}_{i=1}^N$ is the set of all actions performed by each subject i. Actions are assumed to be independent, thus they factorize over trials. To optimize the marginal likelihood with EM, for the E-step at the j^{th} iteration we use the Laplace approximation (Bishop, 2006) to approximate the integral of the marginal (eq. 4.6)

$$P(\mathbf{k}|\mathbf{c}_i) \approx \mathcal{N}\left(\mathbf{k}_i^{(j)}, \Sigma_i^{(j)}\right) \tag{4.7}$$

$$\mathbf{k}_i^{(j)} = \arg\max_{\mathbf{k}} P(\mathbf{c}_i|\mathbf{k}) P(\mathbf{k}|\boldsymbol{\theta}^{(j-1)}) \tag{4.8}$$

and the parameters are estimated at the maximum of the posterior distribution (MAP). For the M-step we estimate the hyperparameters $\boldsymbol{\theta} = (\boldsymbol{\mu}, \mathbf{v}^2)$, by maximising the expectation computed at the E-step, as:

$$\boldsymbol{\mu}^{(k)} = \frac{1}{N} \sum_i \mathbf{k}_i^{(j)} \tag{4.9}$$

$$\left(\mathbf{v}^{(j)}\right)^2 = \frac{1}{N} \sum_i \left[(\mathbf{k}_i^{(j)})^2 + \Sigma_i^{(j)} \right] - \left(\boldsymbol{\mu}^{(j)}\right)^2 \tag{4.10}$$

For the Lookahead model we fitted 1 parameter, for the Discount model 2 parameters, and for the Pruning model 3 parameters. All parameters were transformed before inference to enforce constraints ($\beta \geq 0, 0 \leq \gamma_S, \gamma_G \leq 1$). Specifically, because the optimizing function `fminunc` performs unconstrained optimization we wish to

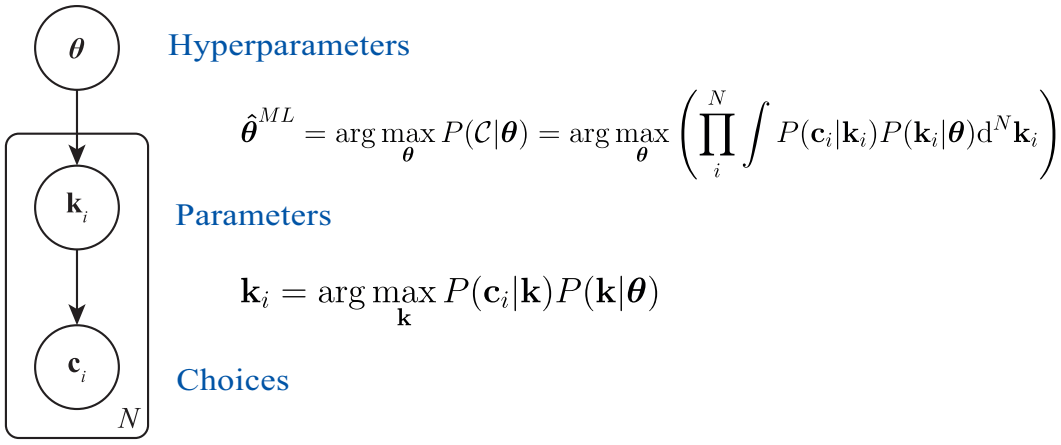


Figure 4.3: Graph for inferring parameters and hyperparameters using the EM algorithm.

transform the parameters in a convenient way in order to preserve the constraints. For example, for the β parameter we performed an exponential transform that ensures the parameter remains positive during inference, and for the different γ parameters we transformed them through a sigmoid function to ensure they take values between 0 and 1.

The above procedures were verified by using simulated data from a known decision process. For this, we simulated the specific task which the human attempts to solve, and we designed an agent that uses the models described in this section to attempt to solve the given tasks. The agent uses one of the model-based models with specified parameters and generates sequences of actions in order to solve the task. Then, we used these sequences to perform inference and recover the specified parameters. With this approach, we can detect possible problems that an algorithm might have. Also, more importantly, we can verify that the model selection procedure (described in next section) can identify from which model the action sequences were generated from.

4.2.3 Model comparison

Given the three models, and given that the models have different numbers of parameters, it is important to compare them to understand which best accounts for the observed data. Following Huys et al. (2012), and having no prior knowledge about the likelihood of each model, we assume that models are equally likely a priori. Thus, we examine only the log likelihood of each model $\log P(\mathcal{C}|\mathcal{M})$. This

quantity can be approximated by the Bayesian Information Criterion (BIC) as:

$$\log P(\mathcal{C}|\mathcal{M}) = \int P(\mathcal{C}|\boldsymbol{\theta})P(\boldsymbol{\theta}|\mathcal{M})d\boldsymbol{\theta} \approx -\frac{1}{2}\mathrm{BIC}_{int} = \log P(\mathcal{C}|\hat{\boldsymbol{\theta}}^{ML}) - \frac{1}{2}|M|\log|\mathcal{C}| \tag{4.11}$$

where $|\mathcal{C}|$ is the total number of choices made by all subjects of the group being examined, and $|\mathcal{M}|$ is the number of prior parameters (i.e., mean and variance for each hyperparameter) that we estimated empirically above. The first term on the right hand side of eq. 4.11 was estimated by standard Monte Carlo approximation. The Bayesian Information Criterion (BIC_{int}) here is not the sum of individual likelihoods, apart from penalizing the model for extra parameters, but the sum of *integrals* over individual parameters thus the *int* (integral) subscript.

$$\log P(\mathcal{C}|\hat{\boldsymbol{\theta}}^{ML}) = \sum_i \log P(\mathbf{c}_i|\mathbf{k})P(\mathbf{k}|\hat{\boldsymbol{\theta}}^{ML})d\mathbf{k} \approx \sum_i \log \frac{1}{N}\sum_{l=1}^{L} P(\mathbf{c}_i|\mathbf{k}^l) \tag{4.12}$$

The second approximation in eq. (4.12) involves a Monte Carlo approximation. In other words, we sample L samples from the empirical prior ($\mathbf{k}^l \sim P(\mathbf{k}|\hat{\boldsymbol{\theta}}^{ML})$), and then average over all $P(\mathbf{c}_i|\mathbf{k}^l)$. With this approach we compare not only how well a model fits the data when its parameters are optimized, but also how well a model fits the data when we use information about where the group level parameters lie on average (Huys et al., 2012).

Although the above gives a good comparative measure of model fit, an absolute measure is needed in order to ensure that the best model does indeed describe the data generation procedure efficiently. Thus, given the MAP estimation of each subject's parameters, we compute the mean total "predictive probability" for subjects N, in a number of trials T, which is the geometric mean of all the $P(c_t|s_t, \mathbf{k}_i)$:

$$\sqrt[NT]{\prod_{i=1}^{N}\prod_{t=1}^{T} P(c_t^{(i)}|s_t, k_i)} \tag{4.13}$$

where c_t is the action selected at trial t by the i^{th} subject, at the state s_t. It is a measure of how probable are the data to be generated from a decision process described by a specific model, characterized by parameters $\mathbf{k}_i$.

4.3 The Tower of London Task (ToL): A Developmental Study

Human planning has been studied extensively using "look-ahead" puzzles (e.g., Klahr and Robinson (1981), Kotovsky et al. (1985), Parrila et al. (1996), Daw et al. (2011), Balaguer et al. (2016)), in which subjects have to pre-plan mentally a sequence of moves in order to transform a starting configuration of a puzzle to a goal configuration, according to a set of rules. In the ToL task (Shallice, 1982), for example, subjects are required to rearrange three balls on three pegs so that the configuration of balls matches a goal state (see fig. 4.4), but in doing so they must adhere to a set of rules or constraints. Thus they must move only one ball at a time, and place it back on a peg before moving another ball. The ToL task can be viewed as a sequential decision making puzzle, with a reward obtained if or when the player achieves the goal state.

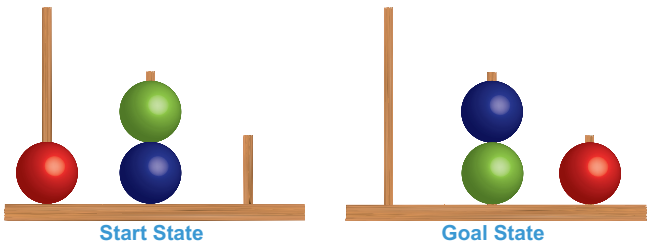


Figure 4.4: A typical Tower of London problem. The task consists of a board with three pegs, each one with different heights, and three different coloured balls. The right peg can contain up to three balls, the middle peg up to two balls and the left one only one ball. The balls are initialy arranged in one configuration on the pegs and the goal is to move balls – one at a time and from peg to peg – in order to achieve the given goal configuration. The problem shown requires 3 moves, but more difficult problems may require up to 7 moves.

The Tower of London task (ToL) was initially developed by Shallice (1982) as a task that would make strong use of the general-purpose planning system (e.g., fig. 4.4). It is thus extremely well-suited for use with computational models to simulate human behavior. Because of the task's suitability in testing planning performance, the task has been extensively used to assess planning abilities of neurologically impaired patients. The particular task is similar to the popular task of Tower of Hanoi (ToH) (Goel and Grafman, 1995; Lucas, 1882) although they differ in structure and rules.

In this section we apply three reinforcement learning models of planning to behavioral data from the Tower of London (ToL) task, and use Bayesian analysis to fit each model. The datasets we used were collected from 3 to 4 year old children and 5 to 6 year old children performing the task and from adults aged >28 years performing a computerized version of the task.

4.3.1 Behavioral Evidence of Planning

Developmental aspects of goal-directed behavior have also been investigated by other researchers. Klahr and Robinson (1981) used a novel variant of the ToH task to assess problem-solving and planning processes in preschool children. They tested 4, 5 and 6-year-old children and found that better performance was observed for tower ending problems[2]. Furthermore, error propensities were related to the age of the child. They argue that although younger children have some of the cognitive capacities for problem-solving processes as in adults, they may differ in encoding and representational abilities.

Bull et al. (2004), in a different study, compared the performance of young children in the ToL and ToH tasks. As the main components in executive functioning are thought to depend upon three core mechanisms (i.e., the inhibition of prepotent responses, shifting, and updating working memory), they investigated possible correlations between the tower tasks and some other specifically chosen tasks that could examine each mechanism separately. They found that age was related to ToL performance; specifically, older children solved more problems correctly. The flexibility to shift among potential moves or subgoals was evident in both tasks and especially in more complex problems. The role of inhibition was found to be more prominent in the ToL task, although they argue that this might be reflective of the different instructional sets governing the two tasks. In this study, it was found that the storage capacity of short-term memory was unrelated to either ToL or ToH performance, consistent with Welsh et al. (1999) and in contrast with researchers who had considered tower tasks to place a substantial load on short term-memory, because of the need to store and retain elements of sequential plan (Pennington et al., 1996).

Tower tasks such as ToL and ToH, are described as higher-order planning tasks

[2]Tower ending problems are problems in which the goal state consists of the three balls stacked at the first peg. In contrast, flat ending problems are problems in which the goal state has all three balls occupying one peg.

because successful completion requires the participant to 'look ahead' and solve the problem mentally before physically interacting with the balls or disks. However, Bull et al. (2004), argued that such tasks need not be solved solely by 'look ahead' planning but also using a real-time, 'perceptual' strategy (e.g., Simon (1975)). Participants may use more direct, on-line processing, where the current tower configuration guides the next move. In other words, the participant attempts to bring the tower configuration successively closer to the goal state with each move, selecting the moves that appear more 'natural' at a given configuration. Perceptual strategies therefore, do not always lead to the shortest solution path and are stimulus-driven, which means that they have little to do with planning. The above findings have important implications for the current study as we were interested in investigating the underlying mechanisms of these types of planning-based and perceptual strategies.

Other evidence that supports the contribution of a non-look-ahead process comes from the work of Goel et al. (1995; 2001). In the first study, patients with lesions in the prefrontal cortex were tested on the ToH puzzle and compared with a healthy control group. Their results suggested that patients' difficulties in solving the task had little to do with planning deficits; rather their performance was affected by an inability to resolve a goal-subgoal conflict, which was indicative of a specific kind of perceptual strategy they used. The patients appeared to employ a perceptual strategy much more than the normal group. In general, both groups seemed to use a general strategy that did not result in the shortest path.

In their second study, Goel et al. (2001) used a computational model to simulate the performance of patients with similar impairments as above and normal subjects. The computational model was built in a hybrid-symbolic-connectionist architecture called 3CAPS (Just and Carpenter, 1992). It was an attempt to model the perceptual strategy described above and managed to capture the main effects of performance in both groups. Their results showed that there is considerable evidence that both control and patient groups were using the perceptual strategy found in their first study, and led them to support the working memory hypothesis of frontal lobe functions. However, this hypothesis was only relevant for a very narrow range of problems, consistent with the findings of Bull et al. (2004), in which working memory did not appear to play a significant role. Bull et al. (2004) suggested that young children did not seem to plan their moves before implementation, as they did not pause before moving the disks/balls (indicating they did not

engage in any planning preparation). Instead they appeared to rely on an online perceptual strategy for deciding their next move in an online way.

In another study, Kaller et al. (2008) tested 4 and 5 year-old children in a variant of the ToL task. They used problems that either required searching ahead for an optimal solution or were solvable by step-by-step forward processing. They found that the 4-year-olds accuracy was lower in problems requiring search ahead strategies, which revealed an age-related effect of search depth, as the older children mastered both types of problems equally well. Initial thinking and movement execution times revealed main effects of goal hierarchy. Goal hierarchy determines the degree to which the sequence of final goal moves can be derived from the configuration of the goal state (Kaller et al., 2004). Search depth affected only initial planning but not movement execution. Furthermore Kaller et al. (2008) demonstrated the importance of problem structure especially when testing sub-populations such as children. Most importantly, though, are their developmental conclusions. The relative inability of younger children in solving three-move ToL problems could be attributed to the failure to look ahead. The observed interaction between search depth and age provides strong evidence for this.

Other studies support the correlation between task performance and age. For example, Albert and Steinberg (2011) explored age differences in strategic planning using the ToL task. Specifically they tested a sample of 890 individuals with ages ranging between 10 and 30 years. On relatively easy problems, mature performance was attained at the age of 17 whereas on more difficult problems performance improved into the early 20s. Their findings also support the claim that late adolescence is a time of continuing improvement in goal-directed behavior.

Our concern in this section is whether intrinsic motivation might play a role in the cognitive processes underlying planning. We use computational methods to explore the effects of more frequent feedback – reflecting intrinsic motivation – on appropriate moves that may guide the subject towards solving the puzzle. This type of strategy is an online strategy, similar to that described above, which guides the subject towards moves that bring the current configuration of the task "closer" to the goal configuration. Specifically, we model an existing dataset from children's planning on the ToL by incorporating a *reward shaping function* (Ng et al., 1999), representing the intrinsic motivation of the child, within the framework of model-based reinforcement learning.

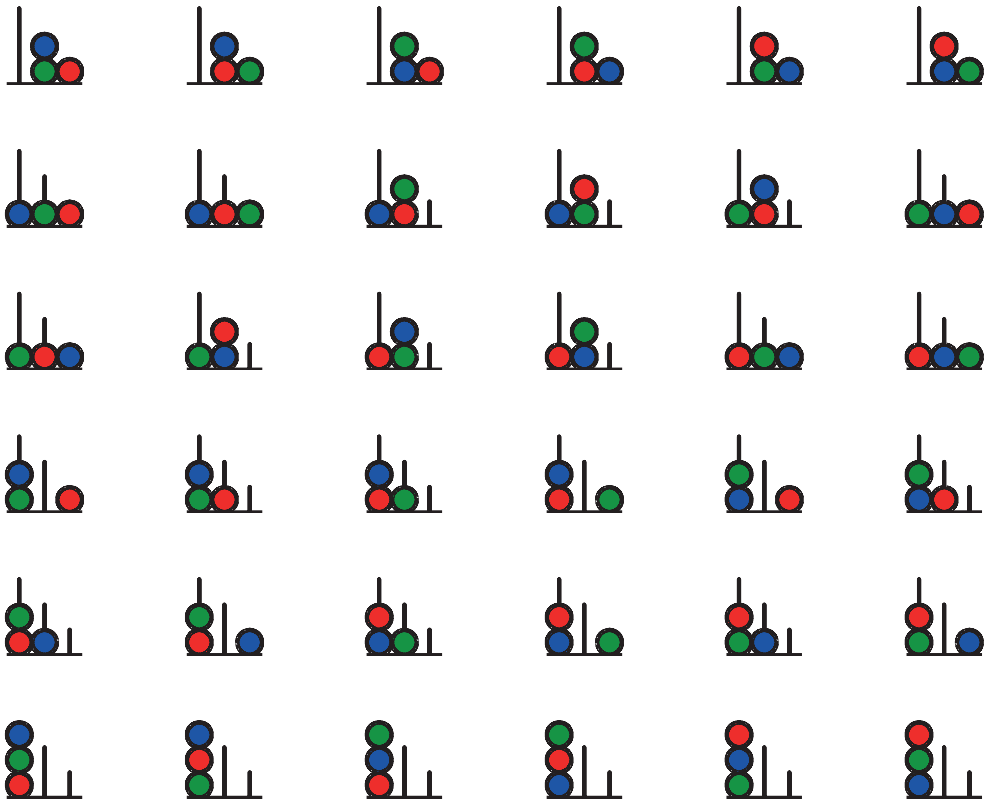


Figure 4.5: Tower of London task state space.

4.3.2 Experimental Procedure

To examine the planning process mechanism, we consider existing data from seventeen 3-to-4-year-olds (mean age 47 months) and seventeen 5-to-6-years-olds (mean age 68 months) on six ToL problems of graded difficulty (Waldau, 1999). The younger children in this study struggled to complete many of the problems, and in both groups some children failed to complete all problems. Therefore, in the analysis below we excluded data from children who were breaking the rules of the game consistently and overall not showing fully understanding of the task, resulting in a final sample of 10 of the younger children and 13 of the older children.

Each child, seated at a table, was shown a physical demonstration of the ToL task in which the experimenter moved balls one at a time to various pegs. The experimenter then asked the child if he or she would like to play the game. After the child agreed, the experimenter started explaining the rules with a parallel demonstration of the task. The experimenter explained that a ball could be picked up only if it was at the top of a peg, could not be placed on the table and then placed at the top of any peg that has room for it. Furthermore, children were instructed to only use one hand for moving the balls and they were asked to decide which hand he or she would use throughout the experimental phase. Next, the child was informed that the aim of the activity was to match the configuration of the game shown in a picture held by the experimenter (which was kept at a visible position throughout the duration of each trial).

After the explanation and physical demonstration of the rules, and upon receiving approval of understanding by the child, the experimenter started a trial phase with 2 to 4 of the 4 problems in fig. 4.6 depending on the pace of learning of the child. This was the first time the child was interacting with the game. It was noticed that younger children had difficulty with inserting balls at the pegs. Many times younger children broke the rules or were unable to continue. The instructor intervened in the solution process to help the child understand the task and the rules. If the child was unable to continue (for instance, did not know what to do next or was too shy, or whatever the case was), the experimenter suggested possible moves in order for the child to understand the goal of the task. Upon completing a task, whether he or she failed or succeeded, the child was asked to shake a small toy rod with a head of an animal at its top, which produced a funny sound. This made the child happy while shaking it and could be perceived as a final reward, so the child would interpret their own self-produced actions as having a meaningful

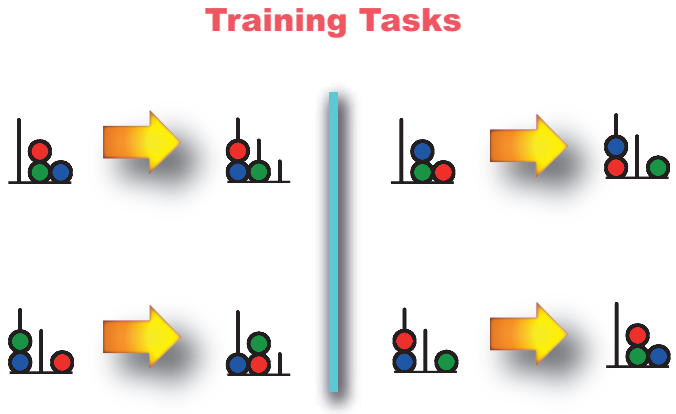


Figure 4.6: Training problems for the ToL.

and rewarding outcome.

When the experimenter was sure that the child understood the rules of the game and the whole process, the child was asked if he or she wished to continue to the real game. Upon confirmation from the child, the real experimental phase began. All children were tested in the problems in fig. 4.7. The data that were used for the model-based analysis were collected from the the child's performance on these 6 problems. Many times, the child initially declared full awareness of the task and the rules, but still often violated the rules of the game. Examples of such violations could be, for instance, picking up a ball in one hand, and moving the balls left to other pegs at convenience while holding the first ball. In these instances, the instructor would ask the child if they were aware that they were breaking rules, but if the child seemed not fully aware of the rules they were allowed to continue playing. With older children the instructor would terminate the trial, after asking the child if what he or she did was allowed and then putting the balls back to the pegs as before the rules were broken, after which the child was allowed to continue from that state.

Given the population and the number of problems, we reanalyzed the original video tapes and obtained 60 and 78 action sequences for younger children and older children respectively. Among younger children, 7 out of 10 performed illegal moves (37 total), whereas 5 out of 13 of the older children used illegal moves (22 total). Illegal moves were counted as the transition of a ball from a peg to the hand and not the other way around. The results are summarized in table 4.1. Some sequence examples from the table 4.1 are given in fig. 4.8.

It is important to stress that the child participant was responsible for completing

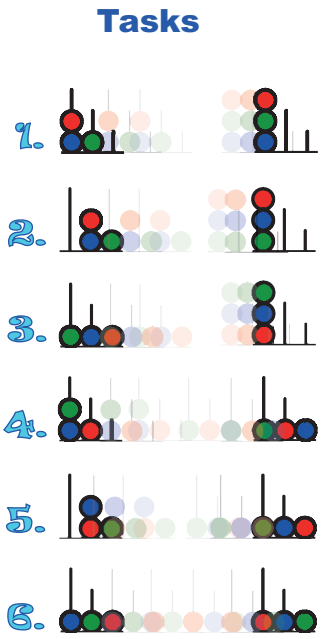


Figure 4.7: Children's ToL Problems. The number of moves needed to solve the problems increases with the number of the problem. Thus, the first one is considered the easiest one whereas the last one the hardest one.

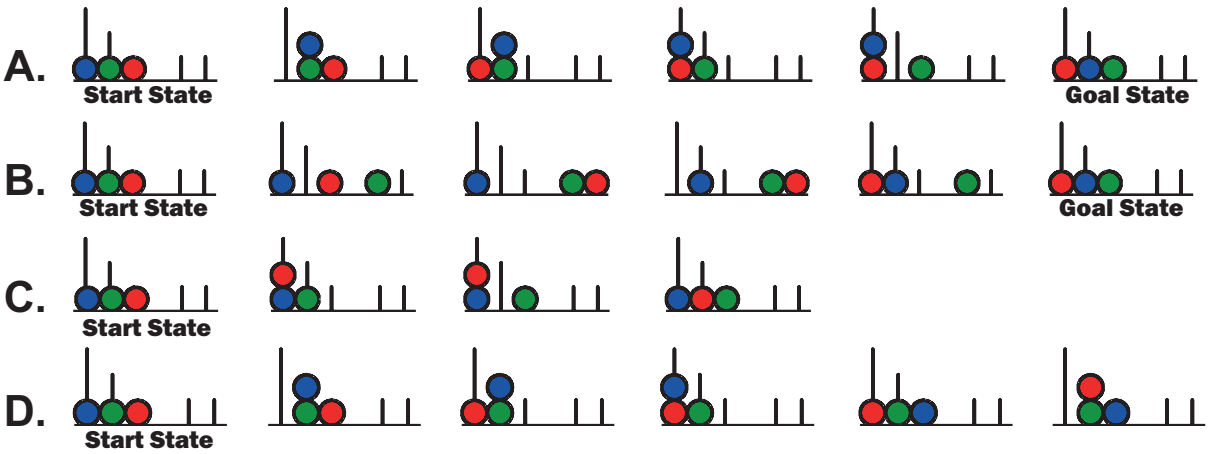


Figure 4.8: Sequence examples of the various cases desribed in table 4.1. **A.** A correct solution of a given task. **B.** Correct solution but with illegal moves. **C.** Perception matching: The subject focuses on producing the configuration of the ToL but ignoring the ball colors. **D.** A failed sequence.

Sequences	3-4yrs	5-6yrs
Reached goal correctly	38.3%	64.1%
Reached goal with illegal moves	40.0%	24.3%
Perception matching	20.0%	5.1%
Interrupted/Stopped	1.7%	6.4%

Table 4.1: Summary Statistics for Children performing in the ToL task. The table shows the percentage of children showing different behaviors in each age group.

a task and declaring that a task was completed. This enables us to capture the "perception matching" effect we described above. The phenomenon of the direction of the behavior towards a perceptual match with the goal state (i.e., reaching a state with the same configuration of the balls as the goal state, except the color of the balls is different in the goal state, and declaring that the goal state reached) is much more evident in younger children. For example, a child might have ended up with the correct flat tower configuration (i.e., one ball at each peg) but with incorrect colors. This phenomenon occurred more often in the younger children and gives us insights about the possible mechanisms that interfere during a planning task.

It is worth noting that in the ToL someone can do location matching without color matching but not color matching without location matching (e.g., the goal state has the red ball in the first peg, second position from bottom to top, and the participant declares as correct position having the red ball at the first peg, but in a different height than that of the goal state). In order to solve the task correctly someone needs to match color, location (peg) and height of the peg. In our study, we mean location matching when we mention perceptual similarity.

Breaking the rules was more often observed in younger children. For instance, although a child might declare full understanding of the rules, the experimental phase seemed to confuse them. As we mentioned, some children's planning process was interrupted by a motor difficulty of inserting the balls into the pegs. In our modeling process,, we attempted to take into account as many of these observed phenomena as possible.

4.3.3 Modelling the ToL task

In problems such as the ToL, the goal is achieved by decomposing it into subgoals and evaluating the order of simple moves towards the goal (Gilhooly et al., 1999). It is this evaluation procedure that guides our approach to planning in such a task.

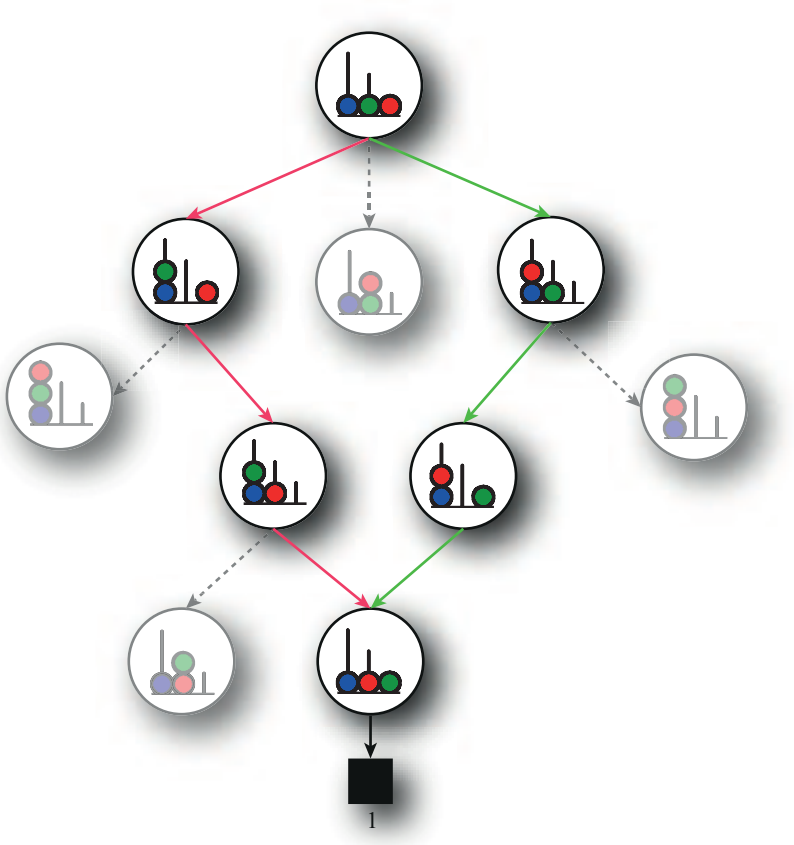


Figure 4.9: A forward internal model that implements planning. The top node is the starting state whereas the bottom node is the goal state, where the subject receives a reward (hypothetically $r = 1$). The two possible solutions are presented: the pink one and the green one. Some connections have been omitted and other alternatives are faded for presentation clarity.

We model children's behaviour on the ToL as a Markov Decision Process (MDP) and follow model-based approaches. In particular, we model the planning process as a forward search model (fig. 4.9). We model planning as an expanding tree with a given horizon. This means that we assume that, for a given moment in the interaction of the subject with the task, and before the subject acts, he or she simulates possible outcomes in the form of a decision tree.

4.3.3.1 The Extended State Space of ToL

Within the empirical study on which this work is based, some children (especially the younger ones) failed to adhere to the task rules. That is, although it was explained to each child that he or she should only move one ball at a time using

only one hand, and although the child in each case claimed to understand this restriction and demonstrated this knowledge in a series of practice trials, sometimes he or she would still pick up one ball in one hand in order to reorder the position of the other two balls that would otherwise require a series of moves.

The above scenario typically occurred when the state of the apparatus almost matched the goal state, with two balls on one peg being in the wrong order (e.g., red immediately above blue when blue should have been immediately above red). One possibility in this case is that the child's look-ahead process suggests to him or her that there is great similarity between the current and goal states, yet any (legal) move would result in a decrease in similarity. From the perspective of search through a decision tree, pruning of the tree might take place when facing such situations, leaving the child with only one viable option – to move both balls at the same time and hence break the task rules.

In order to accommodate breaking the task rules by subjects, we expand both the state space (from 36 states to 114 states) by adding two more locations representing the hands of the child (effectively two additional pegs, each of which can hold at most one ball), and the set of available actions (adding actions corresponding to moving balls to and from the hands). This yields an extended state transition matrix $\mathcal{T} : \mathcal{S} \times \mathcal{A}$ with $[114 \times 25]$ entries.

Thus, for the extended ToL with 25 available actions at each state[3], and a decision tree of depth $D = 3$, the total number of action choices considered by the lookahead model is 16275. This number is large and we reason that children are unlikely to evaluate this number of actions during planning. One possibility is that they prune the decision tree and evaluate action trajectories according to their expected outcome. Therefore, we expect models that prune to fit the data better than simple lookahead models.

4.3.3.2 The Reward Function

From a RL perspective, environmental stimuli combined with external rewards or punishments may elicit certain responses, which ultimately lead to learned behaviours. In this context, extrinsic motivation, which means to be moved to do something because of a specific reward outcome, may be distinguished from intrinsic motivation, which means to be moved to do something because it is inherently

[3]The actual available choices, at each state are given by counting all possible transitions of the balls at the pegs, including the two extra pegs which represent the hands of the child.

enjoyable (Deci and Ryan, 1985).

Intrinsic motivation is evident in animal behaviour, in which it has been found that organisms engage in exploratory, playful and curiosity-driven behaviour even in the absence of an environmental reinforcement (Harlow, 1950). Similarly, researchers in many areas of cognitive science emphasize the importance of intrinsically motivated behaviour for human development and learning. In this section we will describe how intrinsic motivation has a role in the planning process and guides action selection in tasks in which the reward is sparse and received only at the end of these.

The design of the transition matrix is straightforward, as the task is deterministic, but for the reward function further assumptions are necessary. In the Tower of London task, the reward from the environment is given to the subject only at the goal state. In addition to this, however, we assume that subjects are driven step-by-step towards the goal state by an internal reward function, which is related to the similarity of the current configuration of the task, state s_t at time t, to the desired configuration (i.e., the goal state). By "similarity" we mean the degree of overlap, in terms of positions of the balls at the pegs, between two states (as defined in the following paragraph).

We hypothesize that the reward derived from similarity represents the function of intrinsic motivation which guides actions according to what the person believes is good or bad and not by the feedback received from the environment. In other words, in the planning process we assume that subjects evaluate their future actions in terms of not just whether they achieve the goal state, but (for other states) how close they bring them to the goal state. Previous work has shown that such a modification to the reward structure often suffices to render straightforward otherwise intractable learning problems. Indeed, an appropriate modification to the reward function (shaping bonuses) can leave the optimal policy invariant whereas other transformations lead to suboptimal policies (see Ng et al., 1999). In our work we will use shaping bonuses to represent intrinsic motivation and show how it affects the children's planning process.

To calculate state similarity, as required by the internal reward function, we represent each state within the ToL by a set of 24 binary features (bits). For each ball we assign three bits to represent its vertical position on the peg and five bits for the peg that the ball is placed on (three for the real pegs and two representing the hands). According to this scheme if the red ball is at the low-

est position on the first peg then it will be represented as $R_{pos} = (1,0,0)$ and $R_{peg} = (1,0,0,0,0)$. The state vector is the concatenation of the vectors for each ball: $\mathbf{s}_t = (R_{pos}, R_{peg}, G_{pos}, G_{peg}, B_{pos}, B_{peg})$. For example for a given configuration, a state can be represented as: $\mathbf{s}_t = (\textbf{\textit{Red}}, \textbf{\textit{Green}}, \textbf{\textit{Blue}})$, where $\textbf{\textit{Red}} = (\underbrace{1,0,0}_{\text{position}}, \underbrace{0,0,1,0,0}_{\text{peg}})$ $\textbf{\textit{Green}} = (\underbrace{1,0,0}_{\text{position}}, \underbrace{0,1,0,0,0}_{\text{peg}})$ $\textbf{\textit{Blue}} = (\underbrace{1,0,0}_{\text{position}}, \underbrace{1,0,0,0,0}_{\text{peg}})$.

We then define the *similarity between two states* $\mathbf{s}$ and $\mathbf{s_1}$ as the inner product between those states: $\phi(\mathbf{s}) = \mathbf{s}^T\mathbf{s}_1$[4]. In our case all similarities are between a state of interest and the goal state. The reward shaping function bonus therefore has the form $F(s, s_{t+1}) = \phi(\mathbf{s}_{t+1}) - \phi(\mathbf{s})$ where $\mathbf{s}_{t+1}$ is the state one step ahead and $\phi(\mathbf{s}) = \mathbf{s}^T\mathbf{s}_{goal}$ is the similarity between a state and the goal state. The intrinsic reward function becomes:

$$R_{int}(\mathbf{s}_t, \mathbf{s}_{t+1}) = \phi(\mathbf{s}_{t+1}) - \phi(\mathbf{s}_t) = (\mathbf{s}_{t+1} \cdot \mathbf{s}_{goal}) - (\mathbf{s}_t \cdot \mathbf{s}_{goal}) \tag{4.14}$$

According to eq. 4.14, reward is received if the similarity between the future state s_{t+1} and the goal state is greater than the similarity between the current state s_t and the goal state. This means that a child who chooses according to how similar a configuration of the tower is to the goal state will get or feel rewarded. We use the ToL problem in fig 4.4 as an example to illustrate the mechanism of intrinsic motivation, defined in this section, in fig 4.10.

This is an important point: unlike older children and adults, younger children may actually be rewarded during each sub-action, or task step, if the configuration more closely resembles the goal state at a perceptual level. This is in contrast to older children and adults, who might value more the reward at the goal state rather the virtual reward they get if they move to states that resemble more the goal state. Thus, a perceptual strategy ignores more sophisticated planning which is affected by the final reward when the goal state is reached. We expect younger children to be more likely to use such strategies, because we hypothesize that their planning abilities are more restricted than older children. By restricted, we refer to the notion that their planning trees might be over-pruned and thus the available information to plan-ahead is insufficient.

It seems that some children perceive some configurations ("towers" where all three balls are on the longest peg, or "flats" where all three balls are on different pegs) as being the same, independent of the arrangement of colour. The above

[4]With bold letters we denote the state feature vector.

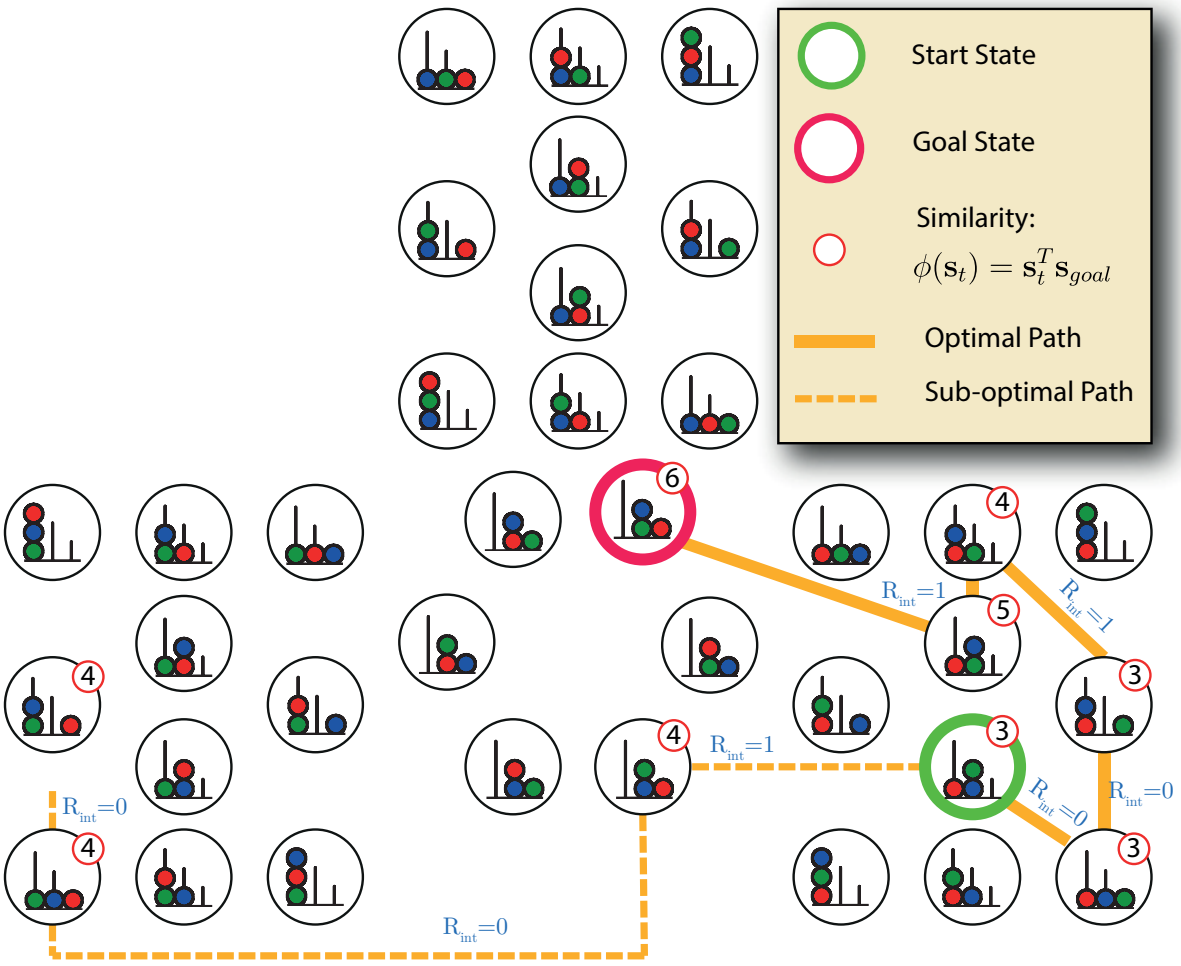


Figure 4.10: In order to solve the ToL task in fig 4.10 with the least possible moves (optimal path), one has to move the green ball first. This move though does not result in a more similar configuration of the ToL to the goal state than the other available options, and thus the internal reward, calculated as in eq. 4.14, is $R_{int} = 0$. On the other hand, by moving the red ball the similarity increases and the internal reward received is $R_{int} = 1$. That specific move is very tempting as the goal state configuration presents the red ball at the third peg, and in the current state the third peg is empty and the red ball is free to move. In such problems the similarity increases early in a sub-optimal solution path whereas in the optimal solution path it increases later. Our similarity formulation captures this tendency to perform actions that lead to states that are more similar to the goal state and affects the planning mechanism.

approach helps us capture similarity in the structure of a particular configuration (i.e., the number of balls on each of the pegs is the same for both configurations

independent of colour). For example, in fig 4.8 **C** the sequence ends into a flat configuration. The similarity of the correct ending configuration in **A** with the wrong one in **C** is 4 (max similarity is 6).

In order to distinguish between goal driven or state-similarity driven behavior, we introduce a weigthing parameter w, and set:

$$R(s,c) = (1-w)R_{goal} + wR_{int} \text{ with } 0 \leq w \leq 1 \tag{4.15}$$

where $R_{int}$is given by eq. 4.14 and $R_{goal} = 1$ if from state s and choice c you reach the goal state, and 0 otherwise. With this definition, eq. 4.5, becomes

$$x = \begin{cases} \gamma_S & \text{if } R_{int}(s,c)\text{is a large dissimilarity} \\ \gamma_G & \text{else} \end{cases} \tag{4.16}$$

This form of reward, weighs the contribution of each type of reward to the total reward function, given a state s and a choice c. Thus, a low w will indicate goal-directed behaviour whereas high w indicates planning driven by state similarity. We hypothesize that younger children will be better modelled by high values of w and older children by low values.

4.3.4 Results and Discussion

We fitted the three models described in section 4.2 to the data, in a way that the extra complexity in the model (extra parameters) reflected better performance of the model in explaining the data. More specifically, we first fit the simplest model and then we repeated the fitting procedure after adding a new parameter and, lastly, comparing the performance of the two. Simple versions of the Lookahead, Discount, and Pruning models were tested with no Similarity function. Their BIC_{int} scores were higher (for example, for the Discount model without[5] a Similarity function the BIC_{int}=1108) than the same models after adding a Similarity function (lower scores indicate better models). Furthermore, an alternative form of Similarity function was tested, which was dependent only on the similarity between the immediate future state and the goal state.

The inferred parameter w was 0.28 and 0.52 (Discount Model estimation) for

[5]In this case there is only one type of rewarded distributed at the end of the task depending on the success of the solution.

Model	Old Ch.	Young Ch.
Lookahead	1441	1066
Discount	1074	797
Pruning	1071	798

Table 4.2: BIC$_{int}$ scores of the three model-based RL models fitted in data from older and younger children.

Parameters	β		γ_G		γ_S		w	
Model	Old	Young	Old	Young	Old	Young	Old	Young
Lookahead	3.59	3.47	-	-	-	-	0.15	0.12
Discount	26.37	28.04	0.69	0.67	-	-	0.28	0.52
Pruning	29.31	30.84	0.71	0.66	0.57	0.57	0.29	0.51

Table 4.3: Mean parameter estimates for the three models.

older and younger children respectively, revealing a significant difference, $t(17) = 3.46$, $p = 0.002$, between the planning mechanisms of the two groups. This suggests, as hypothesized, that younger children pursue a similarity match between goal state and their current state more, whereas the older children demonstrate more goal-directed behavior. By comparing BIC$_{int}$ scores (table 4.2) and mean predictive probabilities calculated using eq. 4.13 (e.g., 5-6yrs old group: Lookahead (0.85), Discount (0.89), Pruning (0.89)), we found that the Discount and Pruning models describe children's behavior better than the Lookahead model, although the extra parameter of the Pruning model does not improve the model predictions beyond that of the Discount model, at least in the specific ToL problems tested here. This may reflect a lack of sophistication in planning ability at these ages. Further investigation of behaviour during specific ToL problems could reveal the importance of various features that affects their planning process, such as the state representation and the Similarity function.

An analysis of choice behaviour according to our models shows that older children prune, in general, more than the younger children. However, the difference is very small. This early termination of the decision tree for the younger participants, appears to be mainly because they are driven by the (perceived) similarity of the current state to the goal state, leading to them 'cheating' by holding two balls at the same time. In addition, younger children tend to overprune their decision tree and mostly are driven by the similarity between states. On the other hand, older children demonstrated a better level of planning (i.e., reaching the goal state by

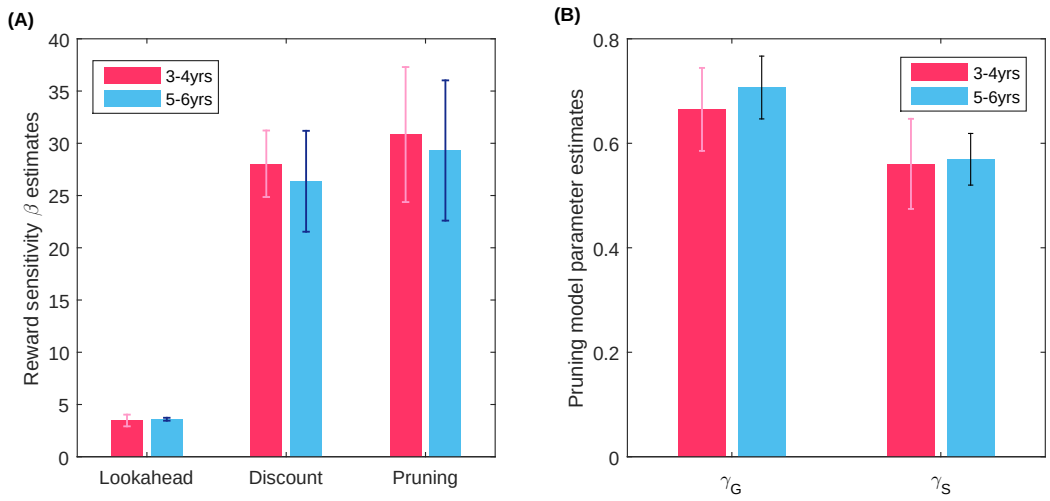


Figure 4.11: Model parameters estimates: (A) Reward sensitivity β mean estimates from the three models. (B) Mean estimates of general pruning γ_G and specific pruning γ_S parameters for the two age groups given the pruning model. Error bars denote the standard deviation from the mean value.

following the rules consistently). They tend to prune but in a way that leads them to the goal state without shortcuts (i.e., without picking up more than one ball at a time). Furthermore, the older children tended not to show confusion in distinguishing very similar states. Finally, looking at the reward sensitivity parameter β (fig. 4.11 **A**), younger children are slightly more greedy in seeking rewards than older children, exploiting internal rewards given by state similarity.

One issue with our approach arises from the choice of the similarity function and the state representation. For example, another alternative more abstract state representation, could take into account only the position of the balls but not the color and vice versa. Different Similarity functions can also be defined with different parametrizations that could potentially better capture the process that directs the tree search toward similar states. In our case we chose a relatively simple similarity function which is based on shaping functions defined by Ng et al. (1999), and the inner product between states. The state representation used, is described in detail in section 4.3.3.2.

This section has demonstrated a method for analyzing human behaviour in puzzle tasks in which the main reward factor is the internal reward, represented by a shaping reward function. By testing it in a real world example, as the above,

useful insights can be gained concerning differences in mental planning between age groups, though further work needs to be conducted to formally explore the relationship between internal reward representations and planning across development.

4.4 The Computerized Version of ToL: An Adult Study

4.4.1 Experimental Procedure

To solve some of the problems that presented themselves in the above experiment with children (such as picking up the balls, rules violation, difficulties to place the balls in the pegs, etc.) an experiment using a computerized version of the ToL task was conducted. 19 adult participants in total aged over 21 years were tested.

The computerized version of ToL was designed in Matlab (MATLAB, 2015) with Psychtoolbox 3.0 (Brainard, 1997; Kleiner et al., 2007; Pelli, 1997). Initially, the participant was seated at a table in front of a laptop screen. Upon agreeing that they were ready to begin the experimental process, the experimenter initiated the ToL script. The script was designed to store actions, completion time for each problem per participant and timing per move execution. Timing was measured from the onset of moving the ball from one peg to another to the moment the ball made contact with the second peg. Time per move is an important indicator of ongoing planning process of the subject (Newman et al., 2003), although these data were not exploited in this study or taken account of in our modeling approach..

The first screen welcomed the participant to the task and explained that she or he would be solving 15 different ToL problems. Participants were instructed to aim to use the fewest moves possible. Next, they entered a training phase with two simple tasks given in fig. 4.12. Here the display split into one large area in which a 2D graphic realization of the ToL was presented, and a smaller marginal area at the left of the screen where a small picture of the goal state configuration of ToL remained on the screen during the trial. A sample display is shown in fig. 4.13. The experimenter explained the rules of the task and allowed the participant to become familiar with the task. In this version, rules were already integrated into the script in order to ensure rule violations were impossible. For example, if the participant attempted an illegal move, the ball would remain at its current location.

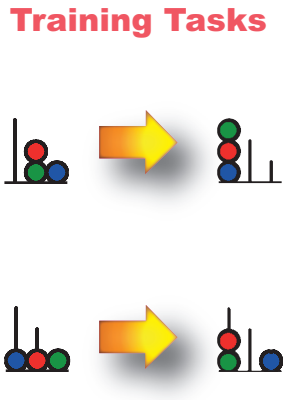


Figure 4.12: Training tasks for the computerized version of ToL.

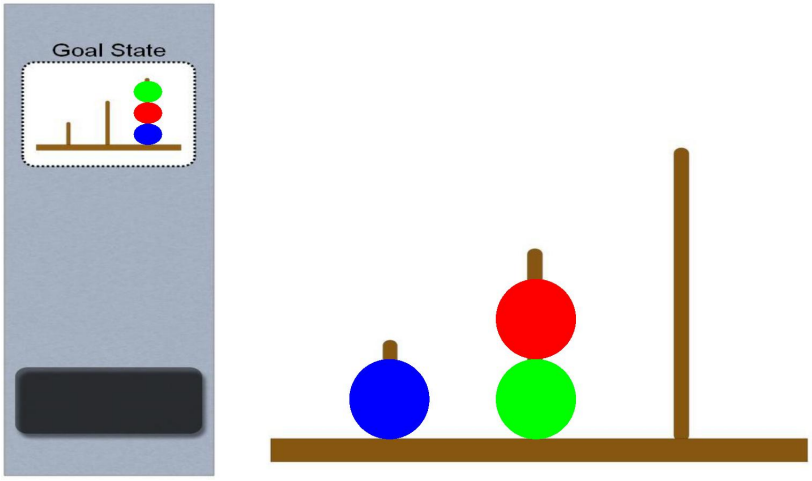


Figure 4.13: A sample screen from the computerized version of ToL. At the top left corner the goal state is presented to the participant. At the bottom left corner, inside the gray box, a 'Next Trial' appears when the participant completes the trial.

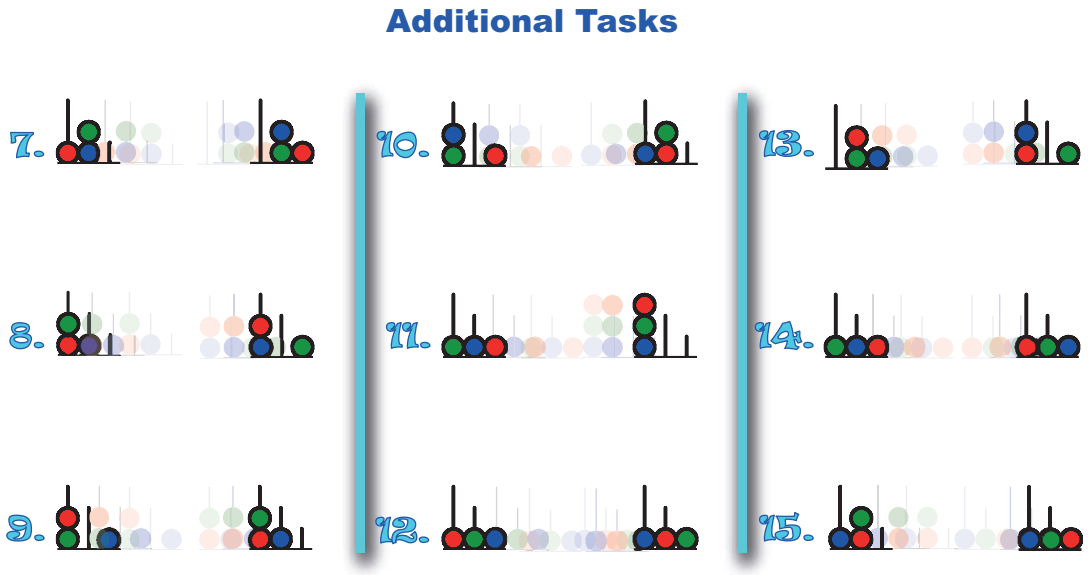


Figure 4.14: The additional ToL tasks for adults. Some tasks were chosen to create the perception that with few moves the configuration of the ToL will be identical to the goal. For example, in problem 8, the first move without planning would be to move the green ball to the 3rd peg as that state is very similar to the goal state. However, this will lead into many more moves than having as subgoal to empty the second peg to help reconfigure the balls at the 1st peg.

When the first trial of the training phase was completed, a box showing "NEXT" appeared on the edge of the screen encouraging the participant to continue to the next problem. The NEXT button appeared only if the task was solved. The first training problems did not require sophisticated planning and the solution was "driven" by the goal state configuration image. The participants did not show any significant difficulty in solving the training trials completely, demonstrating full understanding of the rules and the goal of the task.

After the training phase was finished, a screen appeared indicating that the real experimental phase would begin. Upon agreement, the participant was left alone to complete the tasks. The 15 problems that the participants encountered consisted of the 6 problems in fig. 4.7 with 9 additional problems shown in fig. 4.14. All problems were presented in a random order.

Some problems were chosen to potentially elicit the use of the online perceptual strategy described before. For example, in problem 7, the first move that someone will likely to attempt is to move the red ball to the 3rd peg because this move seems to bring the current configuration closer to the goal configuration. However,

in reality this move actually brings the configuration further from the goal. In order to avoid such mistakes that lead to suboptimal longer paths, the participant must first plan her action sequences several steps in advance. With such problems, we attempted to capture the employment of the perceptual matching strategy that 'encourages' the subject to pursue similarity with the goal states.

4.4.2 Participants

Behavioral data were collected from 19 participants with an age range from 21-45. The participants were mostly highly-educated and drawn from the faculty of various institutes during international conferences that the author attended. Each person was instructed carefully about the rules of the task before participating. As most of the problems were designed for children, they were not very challenging for adults (mean execution time for all the 15 problems was approximately 5-7 mins).

4.4.3 Results and Discussion

The BIC_{int} scores for the three models are given in table table 4.4.Surprisingly, in the computerized version of the ToL task there was no significant improvement with the use of an explicit similarity function. These results might be partially consistent with the online perceptual strategy described in the previous section. It might be the case that, as these subjects were adults, they have mature planning strategies which they used to plan a sequence of moves and then execute them during most of the trials.

Consistent with our previous results, in table 4.4, we observe a high probability for pruning $(1-\gamma_G)$. In addition, the weight parameter w, which denotes the amount of influence that the perceptual similarity has on the planning process, is very low. This accounts for the lack of an improved BIC score with the added similarity. These results are consistent with the hypothesis that adults demonstrate advanced planning processes. However, there is no evidence for the influence of a perceptual strategy. It might be the case that because of the constrained nature of the experiment (i.e., participants were instructed to use the fewest number of moves possible and therefore they needed to plan their course of actions), participants that were keen to use online perceptual strategies were discouraged to do so. In fact, more than one participant complained that planning ahead is not their preferred way of solving the task and that they would have preferred to find the solution by

Model/Parameters	β	γ_G	γ_S	w	BIC_{int}
Lookahead	1.07	-	-	0.09	7421
Discount	1.26	0.10	-	0.06	7427
Pruning	4.22	0.23	0.02	0.76	7428

Table 4.4: Mean parameter estimates for the three models.

'simply playing'.

To further test our models we used a pseudo-R^2 (Camerer and Hua Ho, 1999; Daw et al., 2006) in which we compared the negative log-likelihood obtained by the model with the negative log-likelihood obtained by a null (random) model (i.e., $\beta = 0$, thus the model uses equal probabilities for all available actions at each state). The statistic is computed as $(R-L)/R$ for each subject, with R being the negative log-likelihood of the random model and L the negative log-likelihood of the planning model. The mean value of the pseudo-R^2 for the subjects was 0.98 (compared to a mean value 9359 ± 2559, over 1000 simulations, of the negative log-likelihood of the random model) indicating that the proposed planning model performs better than a random model.

4.5 Application to a Task with Step-by-Step Reward

In the previous sections we investigated human's performance on a task in which they had to reach a goal state where they could get a reward. This group of tasks, in which the reward is given only at the final state, includes the board games such as chess and Go. In some of these games, there is some kind of immediate reward if some particular progress has been made in the game (e.g., specific pieces are captured in chess). However, reaching the goal state has a reward greater than the sum of all the other intermediate rewards and thus all player moves should aim for this. The nature of these games is usually complex and requires careful planning and various strategies to be employed by the player in order to win.

In contrast, there are other planning tasks in which every single action is rewarded or gets punished and the goal is to maximize the total net income. For example, in the stock market the goal is the maximization of the net income at the end of a specified period. The specific trading strategies might be successful sometimes and some other moments unsuccessful. However, what is really impor-

tant is that at the end of the specified period the net income has increased. In this section, we will examine a similar scenario in which participants have to maximize their outcome after a specified number of moves in a small scale'labyrinth.

4.5.1 The Planet Task

To further test planning in such scenarios we employed a task that was used by Tanaka et al. (2006) and Huys et al. (2012) to assess aspects of learning goal-directed behaviors. This task was modified to be suitable for children 3 to 11 year old. It was programmed in Matlab (MATLAB, 2015) with the Psychtoolbox 3.0 (Brainard, 1997; Kleiner et al., 2007; Pelli, 1997). The task consists of a number of states in which there are always two options available. Depending on the state and the selected action, a reward or punishment is given. A participant is asked to choose a course through subsequent states, using a specified number of moves, and to try to acquire the maximum available reward. The task is illustrated in fig. 4.15.

In this task, a participant has to consider not only the reward given after a particular transition, but to plan ahead a path that will lead to the maximum summed outcome at the end of his or her moves. In the ToL task, presented in the previous sections, the reward was given at the end of the task and we examined the mechanisms that potentially affected the planning process. Here, we examine how a human subject, given a number of moves, plans ahead in order to maximize its future outcome considering state-to-state rewards or punishments.

The whole task is presented as a space journey, and navigation through planets in a spacecraft. The states consist of four planets of different colors. Two buttons can be used to make the spacecraft land on a new planet. On each planet the spacecraft collects or loses precious crystals.

From a computational perspective, the task can be modeled as an MDP with the rewards and the transitions considered known, as the subject undertakes extensive training to learn them. The rewards and the transitions can be considered as the model of the environment which participants after training represent as a cognitive map. Eventually, the subject has to plan ahead specific steps evaluating all possible transitions and rewards. Thus, this task can be modeled with exactly the same methods employed in the previous sections.

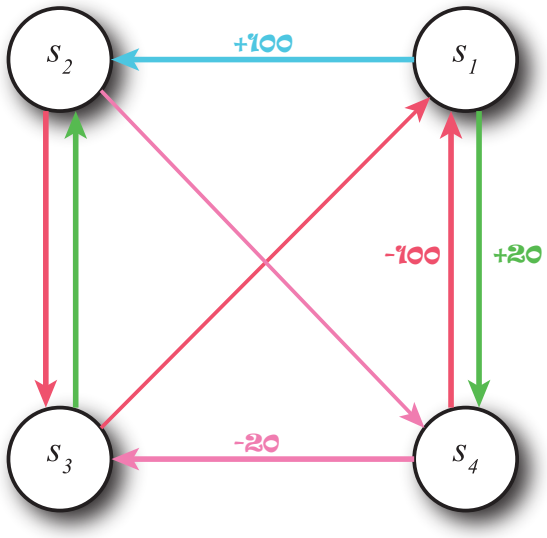


Figure 4.15: Transitions and rewards in the The Planet task.

4.5.2 Experimental Procedure

The youngest participants were recruited and tested by Livia Freier (Ph.D student) at the Center for Brain and Cognitive Development, Birkbeck, University of London. Older children were tested in a non-lab setting (i.e., school). Before the procedure started, the experimenter explained to the parent of the young participant the whole process in detail, while the child interacted with various toys at the reception area of the center. Afterwards, parent and child were led to the lab's area for the testing phase. Younger children were seated on the lap of their parent and were placed in a comfortable position in order to use the laptop. The experimenter initiated the task through the keyboard.

A summary of the whole experimental procedure is illustrated in fig. 4.16. A welcoming screen briefly described the task in few sentences. Then the training phase started. The purpose of this session was for the participant to learn a cognitive map of the two control buttons, indicated by a blue and green sticker at the keyboard of the laptop, and where the spacecraft could land in various cases. The participant could navigate freely to whichever planet he or she wants. At the left of the screen, a map showed the planets and the landing possibilities. The instructor made sure that the child learned successfully how to navigate from planet to planet and assessed the child by asking him/her to move to a planet from the current position. However, this was not the case in younger children, for whom the

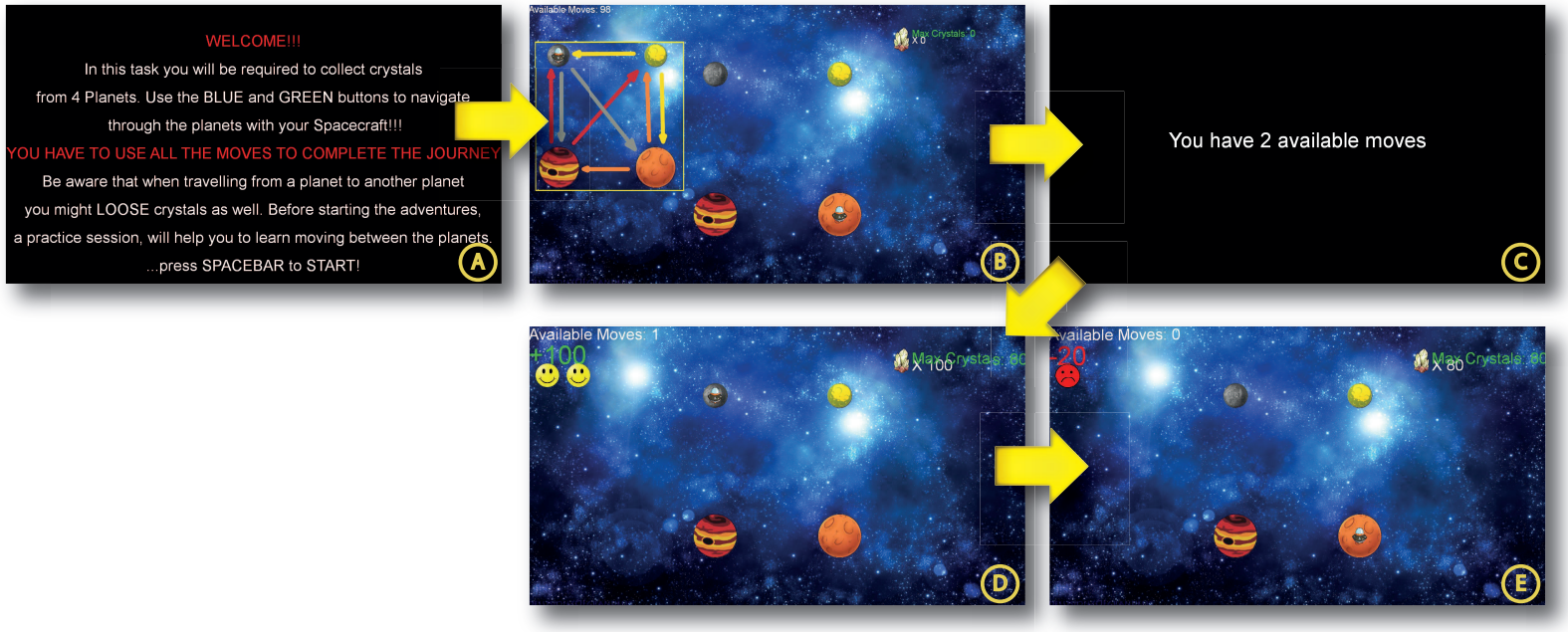


Figure 4.16: The Planet Task. **A.** The welcoming screen and brief instructions. **B.** Training phase. The transitions from state to state are presented on the left and the participant has to learn it in order to proceed to the experimental phase. **C.** Screen that shows the available number of moves before the end of this trial. **D.** A sample screen from the experimental phase in which the participant won the big reward of 100 crystals. **E.** A sample screen from the experimental phase in which the participant loses 20 crystals. Happy and sad faces indicate a bad or good choice.

task eventually was proved too demanding in many aspects.

After the training phase two blocks of trials followed. Specifically, each block differed at the starting position (state 1 and state 2 respectively — s_1 and s_2 in fig. 4.16). In the case that the starting position was state 2, the participant faced two options: one that led to a big penalty and thus would optimally enforce pruning, and another one with a smaller penalty. Although the option of an immediate big penalty seemed repelling, at the end of the course of actions, it led to a greater sum of rewards than the other option with the smaller penalty.

Each one of the two blocks contained three trials, each with a different total number of moves that the participant was asked to use (i.e., 2, 3 and 5 total moves). Each trial of a specific number of moves, was repeated 10 times. There was a maximum reward indicator on the screen, which denoted the maximum possible reward in a specific trial, an indicator for the current collected amount of crystals, and an indicator for the reward that was collected in one transition. The latter one was accompanied by a happy/sad face depending the number of crystals gained.

4.5.3 Results and Discussion

We performed the same analysis as we had done with the previous tasks. We used data from 5 children at the age of 4, 5 children at the age of 5, one child at the age of 9 and one at the age of 11. However, calculating the mean pseudo-R^2 from all subjects, we found a value of $R^2 = 0.03$, a value which indicates that performance was close to the random model. Furthermore, after comparing the mean collected reward from all subjects ($\bar{r}_{subj} = -4363 \pm 1026$) and all trials with the reward obtained by a random model ($\bar{r}_{rand} = -4479 \pm 557$) using a two-tailed t-test, we found no significant difference ($t(11) = 0.40$). The maximum total obtainable reward from the whole task procedure was 800 crystals.

In this experiment we observed various problems. Although our intentions were to design an experimental task that would be feasible for 3 to 11 year old children, younger children did not understand the task well. First of all, as we mentioned, the training phase was not always successful for younger children. Thus, their transition and reward models were incorrect not letting them plan correctly[6]. It seems that, especially for the 3 to 5 year old children, the transitions between states is confusing. Probably, a modification of the task with clear transitions from

[6]It would be very interesting though, to attempt to capture that model instead of assuming a correct model for them during the experimental phase.

state to state, along with control buttons that would represent the direction of movement of the spacecraft, would help them to create a correct transition model. Furthermore, planning steps more than 3 steps ahead seems to increase the load of their planning system. Thus, a block with 1 to 3 available moves might also help them to do the task and eventually test their planning capabilities.

Unfortunately, as it is common with modeling procedures that involve human data, a more careful selection of the population of the participants is needed. In our case, children younger than 5 years old were tested at the Center for Brain and Cognitive Development, Birkbeck, University of London, whereas the rest of the children were tested in schools. The problem of finding schools with children of the desired age for testing contributes a lot to the testing process organization. For example, in our case, the data collected belong to children that come from various age groups making it very difficult to extract developmental results. Taking into consideration all these factors, we could improve the experimental procedures and design better tasks, which will lead to better insights of the process under study.

4.6 Exploratory Work for Future Extensions

In this section, we describe some extensions of the above implementations that might serve as better models to describe the decision making processes involved in the ToL and Planet tasks. These algorithms examine hierarchical aspects of behavior and provide a more realistic framework for planning. We implemented and carry out experiments with all the algorithms described in the next sections in simulated environments. It remains to adapt them to the context of the tasks described above so we can test their suitability.

One of the straightforward extensions to our framework, is to cluster actions into chunks of actions and use an appropriate framework to learn these action sequences and the corresponding policy. This particular policy will account for sequences of actions, instead of one action as in our implementation. The main assumption here is that humans tend to memorize sequence of actions and then use them at will. It is clear that this process consists of an hierarchical approach to a task. For example, to navigate from one room to another in the grid world domain (fig. 2.3), we do not need to consider our actions from tile to tile, if already learned, but rather to directly move to the entrance of the other room. The Hierarchical RL provides such a framework and is described below in detail.

A potential improvement to our planning algorithm is to consider a planning algorithm that does not prune probabilistically. Instead, an algorithm which takes account of the reasons of pruning and integrates them into its functionality, might describe better the human planning process. Monte Carlo Tree Search could be considered as such a model. However, the fitting process to human data of such a model is not straightforward. In the next sections we give a description of the method along with a suitable fitting method.

4.6.1 Hierarchical Reinforcement Learning

Here, we describe an extension of basic RL framework in order to incorporate a hierarchical structure (i.e., divisibility of ongoing behaviour into discrete tasks, each of which are comprised of subtask sequences, and which in turn are built of simple actions). The hierarchical aspect of RL consists of extending the usual notion of action to include whole sequences of actions (Botvinick et al., 2009) which are called *macro-actions*. The *Options* framework formalizes the RL problem with such actions, which can be solved with usual RL techniques. An option is in a sense a 'mini-policy'. Once an option is selected, actions are selected based on that option's policy until the option terminates (Botvinick et al., 2009). Examples of the options framework are: traveling to a distant place, planning a trip, etc. Each of these requires completing subtasks using a sequence of actions. Each step of planning involves foresight and decision making, all the way down to the smallest of actions. Furthermore many of these options might be part of other options thus forming hierarchical structures.

There are two main computational approaches to HRL: Sutton (1988) and Dietterich (1998). For our experiments we used the one with the options framework, following Sutton. According to this, the MDPs are generalized as Semi-MDPs, a special kind of MDPs used for modeling continuous-time discrete-event systems (e.g., see Puterman (1994)). An SMDP is nothing else than an MDP with options. In the four-room grid world, for example, an option could be the transition from one room to another, instead of moving cell to cell.

The RL architecture that was used in HRL is the Actor-Critic, because it is considered to be a suitable model for brain areas such as frontal cortex and the Basal Ganglia. In terms of learning option-specific behavior, the agent has to learn

some predefined subtasks, i.e., to reach a subgoal state[7].

The learning procedure is similar to the one used by the ordinary Actor-Critic. However, in this case the critic, which is responsible for maintaining value functions, maintains not just its usual value function but also a set of option-specific functions. At each step, a prediction error is computed based on the option-specific values of the states visited and the reward received. This prediction error is then used to update option-action strengths (weighted associations from states to actions) which iteratively lead to a behavior with increasing directness toward the options subgoals.

As with the RL framework, the HRL framework has a lot of implications for neuroscience and psychology. Many studies in the past (Cooper and Shallice, 2000; Lashley, 1951) have asserted that the sequencing of low-level actions requires higher-level representations. Furthermore, a number of studies (Bruner, 1973; Fischer, 1980) show that hierarchical behavior is observed through the course of childhood, when simple operations are incorporated into larger wholes. Neural correlates to HRL are tested in Fernández et al. (2010) with encouraging results supporting the hierarchical structure.

4.6.2 Monte Carlo Tree Search Methods (MCTS)

Monte Carlo Tree Search (MCTS) (Kocsis and Szepesvári, 2006) is one of the best-known examples of simulation-based search algorithms. It is a method for finding optimal decisions in a given domain by taking random samples in the decision space and building a search tree according to the results. It has already proved very efficient in Artificial Intelligence (AI) approaches for domains that can be represented as trees of sequential decisions, particularly in games (Branavan et al., 2011; Gelly and Silver, 2011; Heinrich and Silver, 2014) and recently in modeling human decision making processes (Guez et al., 2014b; Hula et al., 2015).

The main idea behind the MCTS algorithm is to simulate thousands of random games from the current position, using self-play. New positions are added to the tree in the form of sub-trees and each node of the tree contains a value that predicts win or loss from that position. This value is simply the average outcome of all simulated games that visit the position. The search tree guides the simulations

[7]The problem of discovering subgoals has been addressed by Kazemitabar and Beigy (2009) and automatic approaches are a topic of much research (e.g., see Bakker and Schmidhuber (2004), McGovern and Barto (2001), Sutton et al. (1999)

along promising paths, by selecting the child node with the highest potential value.

Monte Carlo methods have their origins in statistical physics where they have been used to obtain approximations to intractable integrals. A game-theoretic value of a move, which is an expectation over rewards of that action, can thus be approximated by:

$$Q(s,a) = \frac{1}{N(s,a)} \sum_{i=1}^{N(s)} z_i \delta_s(a) \tag{4.17}$$

where $\delta_s(a)$is the Kronecker delta which is equal to 1 if $a_i = a$ from $s_i = s$, $N(s,a)$ is the number of times action a has been selected from state s, $N(s)$ is the number of times a game has been played out through state s, and z_i is the result (in terms of reward) of the ith simulation played out from s.

It is possible also to improve the reliability of game-theoretic estimates by biasing action selection based on past experience. Moving selection towards certain moves is sensible, after obtaining some experience, as these moves seem to have higher intermediate reward.

4.6.2.1 Bandit-Based Methods

Bandit problems are a class of sequential decision problems, in which one needs to choose among K actions (e.g., the K arms of a multi-armed bandit slot machine) in order to maximize the expected sum by taking the optimal action at every step. The reward distributions are unknown and potential rewards must be estimated based on past observations. This leads to the choice of amount of exploration/exploitation: exploit the action that seems optimal or explore other actions that currently seem sub-optimal but may turn out superior in the long run.

For bandit problems it is useful to know the upper confidence bound (UCB) that any given arm will be optimal. The simplest UCB policy (UCB1, Auer et al. (2002)) dictates to play arm j that maximizes

$$UCB1 = \overline{X_j} + \sqrt{\frac{2 \ln n}{n_j}} \tag{4.18}$$

where $\overline{X_j}$ is the average reward from arm j, n_j is the number of times arm j was played, and n is the overall number of plays so far. The reward term $\overline{X_j}$ encourages the exploitation of higher-reward choices, while the right hand term encourages the exploration of less-visited choices. Bandit theory plays an important role in the

MCTS algorithm as it treats every action as a slot machine and uses a similar action selection method.

4.6.2.2 The Monte Carlo Tree Search algorithm

As we mentioned above, MCTS rests on two fundamental concepts: that the true value of an action may be approximated using random simulation; and that these values may be used efficiently to adjust the policy towards a best-fit strategy. The algorithm progressively builds a game tree, guided by the results of previous explorations of that tree. As the tree is built the values of moves that are maintained in its nodes become more accurate.

The basic algorithm involves an iteratively built tree search until some predefined limit is reached. Each node of the tree represents a state of the domain and directed links to child nodes represent actions leading to subsequent states (in other implementations actions can be considered as nodes as well). The four main steps that are applied per search iteration are:

1. Selection: A child selection policy (tree policy) is followed from the root, in order to find a node with unexpanded children nodes.

2. Expansion: One (or more) child nodes are added to the tree according to the available actions.

3. Simulation: From the new nodes a simulation is run according to a policy (rollout policy) until the end of the game or a terminal state.

4. Backpropagation: The simulation outcome is "backed up" towards the root, updating selected nodes' statistics.

One of the most popular algorithms in the MCTS family is the Upper Confidence Bound for Trees (UCT) algorithm. The goal of the MCTS is to approximate the true game-theoretic value of the actions that may be taken from the current states. To achieve this, a tree is built iteratively. The way that the tree is built depends on the child selection method. The value of a child node is the expected reward approximated by the Monte Carlo simulations. These rewards correspond to random variables with unknown distributions. Thus, we can treat the choice of a child node as a multi-armed bandit problem.

The exploration/exploitation dilemma can be addressed using the UCB1 algorithm in an appropriate form for the tree search problem. Hence, every node

selection is modeled as an independent multi-armed bandit problem. A child node j is selected to maximize

$$UCT = \overline{X_j} + 2C_p\sqrt{\frac{2\ln n}{n_j}} \tag{4.19}$$

where n is the number of times the current (parent) node has been visited, n_j the number of times child j has been visited and $C_p > 0$ is a constant.

4.6.2.3 Cognitive Basis of MCTS

It is worth mentioning that the recent success of Google DeepMind's *AlphaGo* (Silver et al., 2016) in the ancient game of Go, is mainly based on the MCTS algorithm with value function approximation performed by powerful convolutional neural networks. Convolutional neural networks approximate the value function at the nodes of the tree, given the state of the game which is imported as an image to the network, in order for the tree search to use it for planning. These networks implicitly modulate the depth of the tree search, as they can provide an evaluation score of how good is a particular state in terms of the probability of winning the game. Then, the MCTS algorithm will direct the search to the most promising nodes of the tree.

To modulate the breadth of the tree search, Silver et al. (2016) used a policy network (another convolutional network) trained initially with human expert data and later improved by policy gradient learning to maximize the outcome (i.e., winning games). It outputs a probability distribution over available actions from a particular state of the game. In brief, it manages to capture the intuition behind promising moves given a configuration of the board. This type of model is a model-free RL model and is used to approximate a stimulus-response (board configuration-action) function which is used by MCTS to plan accordingly. Perhaps, the human/animal brain might perform similar combination of heuristics and simulated responses which are driven by observed patterns.

The above models, though, do not constitute a straightforward approach in fitting behavioral data. Guez et al. (2013a) successfully demonstrated the use of MCTS in modeling human decision making problem realized in the form of a foraging computer game. Specifically, participants had to control a computer agent to collect tokens scattered in a 24×16 landscape grid. Tokens moved randomly at

various locations at fixed time intervals. A computer-controlled agent woke up at random[8] and chased the player's agent. The participant could escape only at his safe place, located at the bottom left corner of the screen. If caught the agent lost all of the tokens.

Guez et al. (2013a) used various MCTS versions (with rollouts, with value function approximation, etc.). However, such models cannot be fitted with typical Bayesian methods such as Maximum Likelihood to behavioral data as they are characterized as likelihood-free models. Instead, we have to rely on approximate Bayesian computation methods (ABC; for a review see Marin et al. (2012)). Such methods are based on approximating the likelihood function $P(\mathcal{D}|\mathcal{M},\boldsymbol{\theta})$ by simulations (generating data $\hat{\mathcal{D}}$), the outcomes of which are compared with the observed data $\mathcal{D}$.

In these methods, features (summary statistics) should be designed in order to capture the structure and characteristics of the player performance, and therefore simulated data and observed data can ultimately be compared. For example, in the above case, some of the features used were: *Distance from predator*, *distance from nearest wall*, *presence in safe quadrant*, *presence in predator quadrant* and *tokens collected*. In the ABC method, a form of sampling is used to sample a model and parameters ($\mathcal{M},\boldsymbol{\theta} \sim P(\mathcal{M},\boldsymbol{\theta})$) from a prior distribution. Then, with this model governed by its sampled parameters, data $\hat{\mathcal{D}}$ are simulated. Task-related features are computed and the respective results are compared with the features generated by the observed data. Samples are accepted/rejected according to a tolerance criterion. Eventually, the posterior $P(\mathcal{M},\boldsymbol{\theta}|\mathcal{D})$ is approximated and further analysis on the appropriateness of the models can be carried out.

As for all statistical methods, there are positive and negative sides to using ABC-based methods. However, in order for someone to test likelihood-free models in behavioral data this is the way to go. It is obvious that careful consideration should be taken at the experimental design stage. Apart from the suitability of a particular task to test a cognitive function, this task should also be compatible with the feature extraction process. Inappropriate features might lead to information loss and eventually bias the discrimination between models.

The ToL task seems a suitable domain to test human planning with such a MCTS model. The tree search approach used in this chapter assumes that a subject simulates all possible state-visitations but this is not always the case, especially

[8]The probability of waking up was conditioned by three threat levels: low, medium and high.

in tasks where the number of states is large. Furthermore, the pruning process is something that comes naturally from within the MCTS algorithm: the tree is iteratively and asymmetrically generated by the root with longer branches indicating promising and rewarding plan trajectories. The algorithm will not account for all the domain states, rather it will select the ones that lead to some kind of reward. In addition, while the pruning models visit a state and then probabilistically consider/reject the subsequent trajectories, the MCTS algorithm might never simulate a visit to some states, depending on its sophisticated mechanism. This kind of approach is much more efficient in terms of computational cost and might be suitable for explaining the way the human/animal brain approximately operates under planning.

We implemented the MCTS algorithm, but it was used only for simulations in the grid world domain and in the ToL. For using it to fit human data collected from the ToL task, the following summary statistics could be considered: how many times a ball was moved to an empty peg or at the top of another peg; successful solution achieved; optimum solution achieved; 'falling into the perceptual trap' feature which concerns the times someone moves a ball (even if it is not optimum) to the same position and peg as the goal state. It is left for future work to fit different MCTS models to real human data and test its suitability as a model of human decision making.

4.7 General Discussion

Our general goal in this chapter was to shed light on the characteristics of the emergence of goal-directed behavior in development. Considering that, in puzzle tasks, a reward is given only at the end when the task is solved, we investigated what motivates the planning process during each intermediate step of the task. We argued that states similar to the goal state are more likely to be chosen by young children whereas older children are able to plan ahead better.

Linking back to the relevant literature and stressing the importance of a non-look-ahead mechanism, we described a computational framework that accounts for such online perceptual strategies. In particular, it seems that perceptual matching mechanisms must integrate with a general planning process. Therefore, we distinguish a general goal-directed mechanism that drives an agent to a goal, but also a perceptual mechanism that guides action selection by choosing actions that lead to

a configuration that is closer to the goal state. When the perceptual similarity between states and the goal state drops to zero, the online process cannot contribute to the solution of the task anymore and at this point planning takes over.

The similarity function that is utilized by younger children is not monotonic. In some states its value drops to zero and this is where the planning process is initiated. However, older children are not usually influenced as much by states that are similar to the goal, and sometimes they might choose a solution path which features states completely dissimilar with the goal state. These findings suggest that perceptual strategies influence general problem-solving solutions for younger age groups. Moreover, a pre-planning period, followed by plan execution, was more evident in older children and in adults.

Some adults appeared to follow a mixed strategy in which they used classical model-based RL mechanisms when they could not detect any perceptual similarity. These patterns should be investigated more by examining the measured latencies between move execution and overall time to complete a solution to a given problem.

A straightforward extension of these studies could be the investigation of planning strategies following the work of Huys et al. (2015). In this study, the same models were used as in this chapter but with some modifications and extensions. First of all, instead of each model accounting for a single action, it could account for a chunk of actions. This enabled the models to capture frequently used action sequences. Using this type of model that can account for sequences of actions could yield insights into the strategies used.

Further questions occur regarding the emergence of goal-directed behavior, such as how it is shaped from a young age and how it evolves throughout adulthood. Other aspects that we could focus on are to identify the critical ages at which these differences in planning are first observed. Secondly, goal directed behavior, at least in the ToL task, seems to be affected by other elements (e.g., state similarity with the goal state, and features of each state) apart from reaching the goal state. The influence of such elements should be investigated to develop a better understanding of the planning process at different ages. The state representation also plays an important role for planning as features from each state are more or less important in planning processes given the age.

Finally, seeking converging evidence from eye tracking methodologies in addition to the computational models and timing measures could reveal important elements of the aforementioned differences (Hodgson et al., 2000; Kaller et al.,

2009). For example, eye-tracking is an implicit behavioral measure that can reflect implicit anticipations and can thus reveal underlying planning processes, which may not be revealed in overt behavior. In addition, eye-saccades or fixations into specific areas of interest might be related to specific strategies used by humans while solving a planning task.

A computerized version of the ToL, which is much more constrained than the physical, version with alternative problems and difficulties could be used to assess the planning performance of different age-groups. Examining subjects within a broader age spectrum would enable us to test the validity of model-based approaches, address possible modifications and potentially capture developmental differences in the planning process.

The problems in our tasks were selected in order to be feasible but also to challenge even the adults. For instance, some problems were selected in order to give the subject the perception of an easy solution (e.g., fig. 4.4), as the starting state appeared very similar to the goal state, but in reality was not. With such problems we can potentially examine the planning horizon (i.e., search depth) at different ages and to what extent state similarity drives planning. Here, further extensions to the reward function can be implemented and investigated.

At an algorithmic level, dynamic programming approaches (e.g., value iteration) are preferred to model goal-directed behavior as described in Chapter 1. However, the tasks used extensively in the literature (e.g., two state task, referred to Wunderlich et al. (2012)) have a very limited state space. Thus, a dynamic programming algorithm can exhaustively search all possible planning trajectories with low computational cost, given a fixed planning horizon. However, this is not the case in more demanding planning tasks such as the ToL task. Monte Carlo Tree Search (MCTS; Kocsis and Szepesvári (2006)) on the other hand, expands the decision tree, formed by all possible planning trajectories given a planning horizon (i.e., an assumed maximum depth of the decision tree), in an asymmetric way. This means that some states might never be visited. In contrast, in pruning model-based models a state is visited and then the decision tree expansion is terminated in a probabilistic way. Such an approach is more suitable for modeling tasks with a large state space, and MCTS is a promising candidate model for the planning process. Further characteristics of the algorithm, such as the roll-out policy (i.e., taking random actions from a leaf node of the tree till the end of the game or a terminal condition), could be explored and related to the cognitive functions

underlying goal-directed behavior.

Although the performance, in the ToL task, of different aged population samples has been previously examined (Albert and Steinberg, 2011; Anderson et al., 1996; Baughman and Cooper, 2007; Gilhooly et al., 1999; Newman et al., 2003), no links between ToL and model-based Reinforcement Learning approaches have been introduced previously in the literature. We hope that with the above suggestions we have contributed to the theoretical framework involving model-based approaches to planning. In addition, our models could be improved by making use of both behavioral and eye-tracking data. Using different complementary modalities could provide a richer view of the mechanisms underlying the planning process.

4.8 Highlights

In this case study we investigated the mechanism that guides planning in tasks in which the reward is sparse, usually given only at the end, and which are quite challenging in terms of discovering the whole solution path from the beginning. Our main hypothesis stated that cognitive search is affected by intrinsic motivation in order for humans to plan efficiently in such tasks. This was based on the experimental observations that participants plan and make moves at each time step regardless the absence of rewards or points at each time step. The intrinsic motivation was represented as a reward function dependent on contextual features of the task – and affected by the participant's perception system (not examined here). That characteristic was responsible for generating various behavioral patterns found in the experimental results. The task that was chosen was the Tower of London task (ToL).

In order to explain computationally that phenomenon we employed Model-based RL models, as the approach is appropriate for planning tasks. More specifically we fitted **three models** – *Lookahead*, *Discount* and *Pruning* – to data collected in **two** different **studies**: children physically solving ToL problems and adults playing a computerized version of ToL. The data collection from the video tapes, the computerized version of ToL and all the computational work–model design and analysis–were performed by the author. The model fitting was performed with Bayesian methods and the results can be summarized below:

- In tasks such as the ToL, in which the reward is received only when the task is solved, models that take account of perceptual characteristics of the task

(similarity among states) described better the observed patterns of the participants playing the puzzle rather than models which consider only the final reward. The intrinsic reward we introduced was represented as an additional state-by-state reward function and was dependent on the similarity of the goal state and the states encountered as options at each state of the puzzle.

- Younger children are affected more by the perceptual similarity between their current state and the available future states in their decision tree as can bee seen from the value of the w parameter.

- Older children are more likely to prune their decision tree as it was reflected in the inferred parameter γ_G which represents the probability of terminating the search tree. This indicates a more developed planning-decision system compared to younger children.

- The reward sensitivity β for younger children was slightly higher than for older ones indicating greedier behavior in seeking rewards compared to older children, and exploiting internal rewards given by state similarity

In the computerized version of the ToL we argue that state similarity was not important for planning performance. This outcome was expected as the participants of this experiment were adults and were strictly instructed to plan their moves before acting.

Apart from the experimental analysis there was a significant amount of computational work that considered other methods such as Monte-Carlo Tree Search and Hierarchical Reinforcement Learning as potential candidate models for modeling planning. The work was focused on simulations rather than model fitting.

Lastly, we attempted to design a computational experiment to test planning abilities and performance of young children. Although model fitting did not give meaningful results for reasons discussed in the main text, we gained useful insights into computational experimental design for that age.

Chapter 5

Learning of Causal Relationships Between Continuous Human Actions

ABSTRACT

In this chapter, we use computational methods to describe the mechanism with which infants and adults employ in order to learn causal relationships between actions and effects. Our efforts are based on experimental evidence from eye-tracking and behavioral data that infants and adults can learn the underlying relationship between actions and effects during a demonstration phase, and transfer that knowledge when they attempt to generate the observed effects. To explain computationally the observed patterns we mainly rely on model-free RL methods as these are suitable for learning by trial-and-error. Furthermore, we introduce a theoretical framework based on Bayesian adaptive planning that can effectively describe the whole process of learning by demonstration and transfer the acquired knowledge to the actual interaction with the demonstrated task.

5.1 Introduction

Humans receive on a daily basis a continuous stream of information in the form of sequential multimodal sensory stimuli (images, sounds, etc.). Causal variables, that provide structure in this flow of data, are embedded within this temporal stream of events. For example, when observing a person making a cake, we receive a series of images containing actions and sounds. This sensory information contains a structure: the actor always first gathers ingredients, prepares them, mixes them

and eventually places them in a cake tin. These actions can be grouped into a hierarchy of macro-actions (i.e., ingredient mixing). Within a group, the order of each action might not matter (i.e., mixing an ingredient with another one). However, other actions (first pouring the batter into the cake tin before baking the cake) must occur in a specific order for the outcome to be successful, and they reveal a causal relationship between actions and their order. In our example, the baking action results in the ready-to-be-served cake.

Social reasoning (and especially causal inference) depends on understanding the relationship between actions, goals and outcomes. Causal inference refers to the challenge of identifying the subsequences within a stream of actions or events that correspond to the appropriate causal relationships between these events and their outcomes. Causal inference can result in learning and imitation of particular actions (Buchsbaum et al., 2015), which precedes goal identification and selection for the actor, and finally results in the ability to use this learned sequence of actions (via observation) to achieve his or her goal. Adults have acquired a lifetime of experiences that facilitate accurate and rapid causal inference. However, for naive infants, the statistical information contained within stimuli streams provides a crucial source of potential knowledge and learning. The learning processes taking place during observation of action sequences, and how young infants use them during online processing, is still open to investigation and can lead to significant contributions to developmental sciences.

There is a large body of evidence suggesting that humans can use statistical patterns in spoken languages to segment words from continuous speech from early in development. Saffran et al. (1996) reported that 8-month-olds can segment a continuous stream of speech syllables into word-like units. In their classic paradigm, they created two different artificial languages by combining 12 different syllables to form four trisyllabic words, with no specific meaning, for each language. A synthesizer generated a continuous stream without pauses of randomly ordered words from one of the two artificial languages. Critically, after a 2-minute exposure to this stream, infants were tested by exposing them to intact 'words' versus 'part-words', which included syllables spanning the boundaries of different words and thus featured lower transitional probabilities between syllables than the intact words. Their measure of learning was whether infants increased their gaze duration towards the stimulus when listening to part-words relative to words, which is considered an indication of recognition of an unfamiliar or novel stimulus.

Aslin et al. (1998) studied in detail the statistical computation used by infants to solve this word-segmentation task in order to identify the specific learning mechanism used. It was found the infants can discriminate the differences in transitional probabilities between words and part-words. The model used was based on the definition of the conditional probability

$$P(B|A) = \frac{P(A,B)}{P(A)} \tag{5.1}$$

where A and B are successive syllables. It is worth noting here the comment of Aslin et al. (1998), regarding the importance of computation of conditional probabilities:

> "The computation of conditional probabilities is an important ability because, in language as in many other patterned domains, relative frequency (even complex frequency, such as the frequency of co-occurrence of pairs or triples of items) is not the best indicator of structure. Instead, significant structure is typically most sharply revealed by the statistical predictiveness among items (i.e., frequency of co-occurrence normalized for frequency of the individual components; see Rescorla (1966))" (Aslin et al., 1998, p. 323).

In a footnote following this comment the authors stated:

> "Rescorla (1966) showed that classical conditioning in dogs involves the computation of a conditional probability or correlation between a tone and subsequent presentation of shock. One might ask, then, if human infants can show classical conditioning, is it not already known that they can compute conditional probabilities? In fact, to our knowledge, Rescorla's paradigm has not been run with human infants. But, more important, our own task involves quite a different order of magnitude of processing than Rescorla's. Our word segmentation task, if performed in its entirety, involves the on-line (running) computation of 20 different conditional probabilities, each over 45 to 90 occurrences of the component syllables and 9 to 90 occurrences of syllable pairs, during a 3-min learning period. Eight of these 20 conditional probabilities are included in our test items. Our study, thus asks not merely whether infants can compute a single conditional probability, but whether they can compute a large number of such probabilities simultaneously" (Aslin et al.,

1998, p. 323).

An important element of causal inference, is action segmentation: dividing a sequence of actions into shorter sequences, or individual actions, and determining which ones of those can lead to effects in the world. Baldwin et al. (2008) demonstrated that sensitivity to statistical regularities in continuous action sequences is also evident in humans. In other words, if people can distinguish words from an artificial language as described above, they can also extract action steps from within continuous sequences which have specific causal outcomes, and they can learn these corresponding associations.

Buchsbaum et al. (2015) investigated thoroughly various scenarios of action segmentation within continuous action sequences. In a series of experiments, they presented actions in a continuous way, and their first experiment showed that the continuous boundary judgment measures, used during event segmentation, align with the sequence discrimination measures used in statistical segmentation research (Baldwin et al., 2008; Meyer et al., 2011, 2010). In a second experiment, they found that people experience the action subsequences that they extract from continuous streams as meaningful and causal sequences. Lastly, their third experiment showed that people were able to extract the correct causal variables from within long action sequences. They also found causal sequences to be more coherent and meaningful than other sequences with equivalent structure.

Kidd et al. (2012) used a paradigm adapted from Saffran et al. (1996), using actions and objects in their stimuli sequences. They assumed that infants maintain expectations over observed events according to the observed frequencies of these events. According to this, infants should create a representation of their guess (i.e., a probability θ_i, where $i = 1, \ldots, N$ with N the number of events) of the true distribution of events, which is based on the number of occurrence of each event, and maintain a belief over these representations.

To test their hypothesis they used a a Dirichlet-Multinomial[1] model which took as inputs a sequence of observed events or transitions between events, to compute expectations about which event is more likely to occur in the future. They argued that their results showed that human infants employ such an inferential process for

[1]They were interested in estimating a multinomial distribution parametrized by $\boldsymbol{\theta}$, where θ_i is the true (unobserved) probability of an event. This event has a multinomial likelihood, and as it is common in Bayesian statistics, its conjugate Dirichlet distribution was used as the prior distribution for the Bayesian inference.

learning about events in the world. In a following study (Kidd et al., 2014), they found similar results by using a sequential auditory stimuli paradigm.

In our work, we aim to investigate the role of learning causal effects in a sequence of actions. We attempt to answer the following questions: Are infants able to detect and extract the statistical structure within a continuous dynamic stream of motion? Does a salient effect after a specific order of actions provide a reinforcing cue that enables naive infants to predict these actions more accurately? Using computational models with eye-tracking and behavioral data from a novel experimental task, we give useful insights of the learning mechanisms that take place and attempt to give explanations of the above phenomena.

5.2 Materials and Methods

5.2.1 Experimental Task

The experimental task that this chapter models, was designed and developed by Monroy et al. (2015a; 2015b), who also collected the data. We give a description here for completeness (further details are reported in the cited manuscript). Adult and infant participants observed a video of a sequence of action events using a multi-object toy. This toy featured six objects that afforded distinct actions and a central star-shaped light (see fig. 5.1A). The same toy used during the video was then presented to participants during a post-video behavior play session.

Four sequences of 96 total actions were constructed, using the program "Mix" (van Casteren and Davis, 2006). These sequences were constrained such that two deterministic pairs occurred 12 times each (example: A-B and C-D), while all other possible pairs occurred with equal frequency and thus featured transitional probabilities of 0.167 (see fig. 5.1B). One deterministic pair caused the central star on the toy stimulus to light up (the 'Effect' pair), while the second deterministic pair resulted in no effect (the 'Non-effect' pair). The second actions of each deterministic pair were labeled *targets*. The effect onset occurred at a natural mid-point of the target action and the offset occurred after the action ended. For example, during the target action *open*, the light turned on the moment the yellow door was fully open and turned off again after it closed[2].

[2]For simplicity, throughout the rest of this chapter we refer to the first action of a pair (Effect or Non-effect) as Action 1, and to the second action of a pair as Action 2.

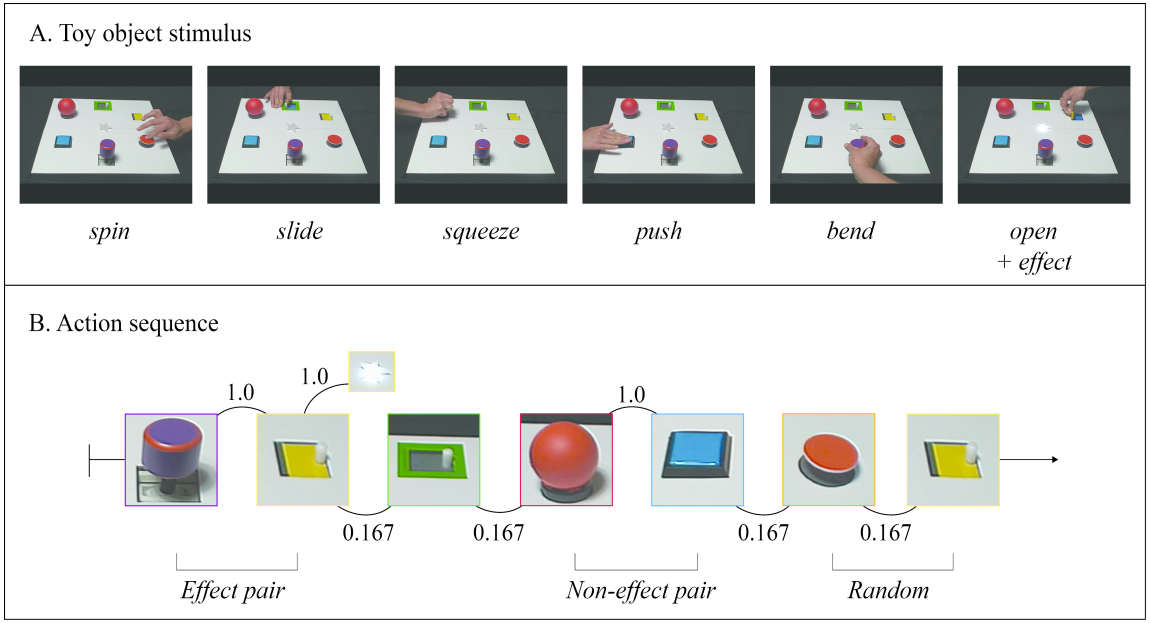


Figure 5.1: Action pair types and example frames (TP= transitional probability). Source: Monroy et al. (2015a; 2015b). Used with kind permission of Claire Monroy.

We emphasize that the actions observed are not continuous, but rather discrete, as the actor performs one action (i.e., *bend*) and then disappears from the screen for one second before reappearing to perform the next action. However, this brief pause between actions preserved the natural timing aspect of the actors' movements, as it reflected the amount of time required for the actor's hand to return towards her body, change direction, and move towards the next object that would be manipulated. This approach also provided a time window during which no movement was occurring on-screen, to allow infants to make predictive eye movements towards the object they expected would be the next target.

5.2.2 Experimental Procedure

Data was collected from one group of adult participants (N=50, mean age=20.7 years, range: 18-25, SD=2.29) and one group of infant participants (N=53, mean age=19.25 months, range: 18.5-20.5 months, SD=2.38, 22 females). All participants were seated on a chair in front of a Tobii eye-tracker presentation screen. Infant participants were seated on a parent or caretaker's lap during both phases of the experiment. Prior to the action observation phase, eye gaze was calibrated using an age-specific calibration sequence[3]. Following successful calibration, participants were shown one of the four possible stimulus sequences. These sequences contain 96 actions performed on the toy-box by an actor (only the actor's hand is visible on the screen). In all sequences the action transition between the actions of the Effect-pair or Non-Effect pair appear with probability one.

After the video presentation, participants then freely interacted with the toy stimulus (one minute for adults and three minutes for infants or until they became disengaged). Importantly, participants were given no explicit task: adults were simply told they had one minute to play however they like with the object, while infants were simply presented with the object on the table before them. The experimenter monitored the participants' behaviors and pressed the light button (the participant could not see that action) if the participant performed the Effect pair. This session was videotaped, and the first two minutes of the infant's behavior were later coded offline to assess action performance.

[3]Calibration procedures differ slightly for infant and adult populations. For complete details, refer to the cited manuscripts.

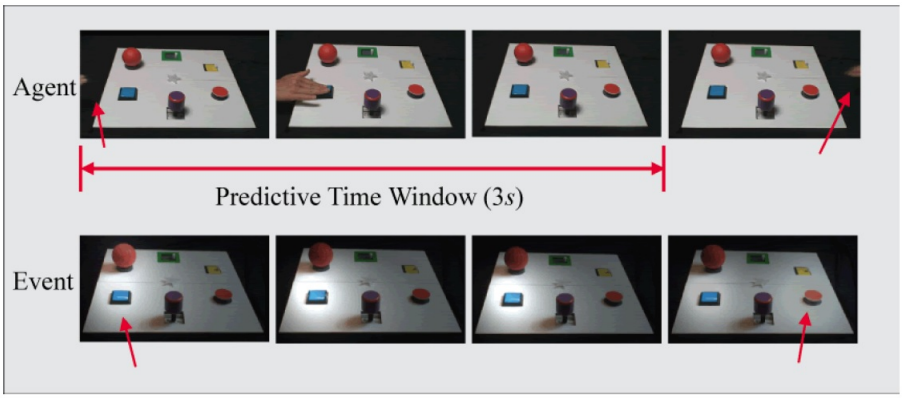


Figure 5.2: Predictive time window. Example frames illustrating the predictive time window from the learning videos. Red arrows indicate the first frame in which the hand appears. Source: Monroy et al. (2015a; 2015b). Used with kind permission of Claire Monroy.

5.3 Data Analysis and Results

5.3.1 Eye Movements data

Raw data from the eye-tracker was preprocessed (see Monroy et al. (2015a; 2015b)) and imported into MATLAB for further analysis. Areas of interest (AOI) were defined around each object as well as the light. Monroy and colleagues were interested in participants' abilities to predict the target actions (i.e., the second actions of deterministic pairs) before they occurred, which they considered a measure of learning the action structure. These authors defined a fixation as *predictive* if it occurred in the time window (fig. 5.2) during which the agent was performing the first action of a pair. These actions provide the observer with the necessary information to make an accurate prediction about what will occur next in the sequence before it actually occurs.

Monroy et al. (2015a; 2015b) examined participants' fixations to the correct target location within each predictive time window. Fixations to the AOI of the target action were counted as correct, while fixations to any other AOI were counted as incorrect. The AOI corresponding to the object currently being manipulated or moving was always excluded from calculations. For the Effect pair, correct fixations also included fixations to the star. Fixations to the action effect were also excluded from calculations for the Non-effect pair. For each pair type, we calculated the proportion of correct fixations out of the total fixations to all objects (fig. 5.3).

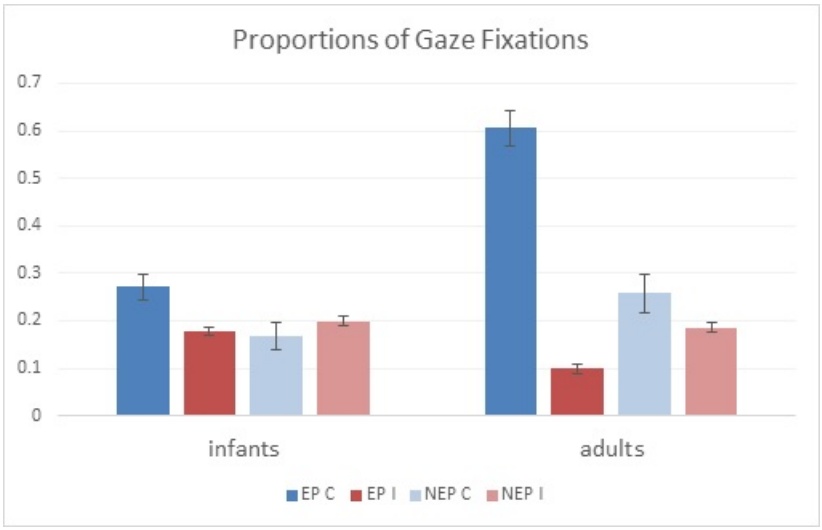


Figure 5.3: Proportions of fixations to correct (C; blue bars) or incorrect (I; red bars) locations for the Effect pair (EP; left) and the Non-effect pair (NEP; right) for each age group.

The incorrect proportion was defined as the average number of fixations to the four remaining objects, out of the total number of fixations to all AOIs (Tummeltshammer and Kirkham, 2013). This location measure represents a preference for looking toward the correct target action, relative to other actions, before it actually occurred. For the Effect pair the proportions are:

$$Correct = \frac{\text{number of looks to target+star}}{\text{total number of looks to all AOIs}} \tag{5.2}$$

$$Incorrect = \frac{\text{number of looks to other 4 objects/4}}{\text{total number of looks to all AOIs}} \tag{5.3}$$

and for the Non effect pair:

$$Correct = \frac{\text{number of looks to target}}{\text{total number of looks to all AOIs}} \tag{5.4}$$

$$Incorrect = \frac{\text{number of looks to other 4 objects/4}}{\text{total number of looks to all AOIs}} \tag{5.5}$$

As it can be seen in fig. 5.3, the results from the data analyses showed indications of learning in both infant and adult age group. Adults made significantly more correct than incorrect fixations across pairs (see fig. 5.3, right histogram), which

was confirmed by a 2 (Location: Correct vs Incorrect) x 2 (Pair: Effect vs. Non-effect) repeated-measures ANOVA test. This test yielded a main effect of Location, such that the proportion of fixations to the correct location was significantly higher than to incorrect locations, $F(1,41) = 12.18$, $p = .001$, $\eta_p^2 = .23$. Further, there was a Location x Pair interaction, such that the Effect pair elicited a greater difference between correct and incorrect fixations than the Non-effect pair, $F(1,41) = 69.05$, $p < .001$, $\eta_p^2 = .63$. The same repeated-measures analysis with the data from the infants also showed main effects of Location and Pair, and a Location $\times$ Pair interaction. Infants made a significantly higher proportion of correct than incorrect fixations across pairs. Further, as with the adults, pairwise comparisons revealed that the difference between correct and incorrect fixations was significant for the Effect pair but not for the Non-effect pair (refer to the manuscripts for a detailed analysis).

In sum, these analyses revealed that both infants and adults demonstrated in their gaze behavior that they were able to detect the statistical regularities of the pair structure, and make correct predictions towards upcoming actions when they were provided with sufficient information (i.e., after observing deterministic transitions between the actions comprising a pair).

5.4 Bayes-Adaptive Markov Decision Processes

5.4.1 Introduction

We stress that the above task was designed to address the problem of spontaneous learning from action observation. There was no indication that a particular goal should be pursued by the participant. In this section we describe our first approach to model the whole learning process, given the predefined-pretested task and the subjects' data.

The toy-box task was designed to test the ability of human participants to detect structure in other people's behavior, and more specifically, whether humans use this information to guide predictive behaviors and their own action choices. The combination of the two phases (action observation and action selection) reflect two distinct forms of behavioral evidence for whether participants did indeed discover some kind of structure in the sequential actions of the actor. Further, the behavioral evidence indicates that these two phases were related to each other, which suggests

that participants may also transfer the knowledge acquired during the observation phase into their own action selection.

We now turn to the first modeling approach that we implemented in an effort to formalize computationally the mechanisms by which participants may achieve this transfer of information. In order to explain our first approach, we illustrate the process for solving a kind of a riddle: "*From the sequences demonstrated to you, which one turns on the light?*". For example, to turn on the light, a specific sequence of moves should be executed (Effect pair). The correct sequence is demonstrated in the video but it is not straightforward to extract it. We assumed that the subject has an initially (incorrect) model in her mind of how the light is turned on. Then according to what she infers from the actor's action sequences, she will update her model to an approximation close to the real underlying model which describes the mechanism of turning on the light. Below we introduce a possible framework relevant for this reasoning, using well-studied models in Bayesian RL (Guez et al., 2014a, 2012).

The interaction of an agent with the real world, in situations in which he or she makes a decision in order to maximize the expected future reward - which can be described as a Markov Decision Process (MDP) - can be decomposed into two closely-related 'information-exchange' processes: *Simulation* and *Interaction* with the real world. During each process, the agent samples the corresponding experience: in the Simulation phase, simulation-based experience is sampled from an internal model which approximates the real MDP; during the Interaction phase real experience is sampled from the environment (the real MDP). These two processes could be described algorithmically by the integrative architecture of Planning and Learning, Dyna, discussed in Chapter 2.

As we described in Chapter 4, planning involves simulation of possible action sequences in a form of a tree search (for example see Daw et al. (2011); Dolan and Dayan (2013); Huys et al. (2012)), under transition and reward dynamics defined by a model of the environment. The Bayes-adaptive Markov Decision Process (BAMDP) framework allows us to consider all possible models. For a task such as the toy-box, we assume that the participant has to infer the correct transitions that switch on the light. In this case, it is possible that the participant has a current belief about which actions change the state of the toy-box (i.e., switching on/off the light). This can be an instance of an MDP. After the participant starts interacting with the toy-box, then the belief about the dynamics of the task changes and, thus,

in the Simulation phase, he or she, will use a different MDP reflecting the newly acquired information about actions and effects.

In this section we use the BAMDP framework to model the planning process that takes place in a task such as the toy-box. First, we describe the main mathematical framework and then we describe how this framework can be applied to: a) use prior information from the video demonstration phase, and b) combine that information with the different hypotheses a participant might have over the state transitions of the toy-box.

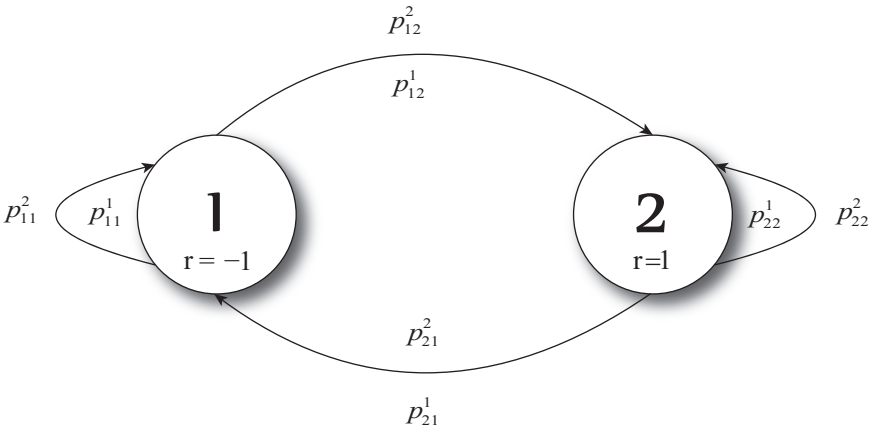


Figure 5.4: A simple BAMDP. The $p^a_{ss'}$ denotes the probability of transitioning from state s to state s' by choosing action a. In this BAMPD we have two states where a reward is received upon reaching each one of them (i.e., $r^a_{s1} = -1$ and $r^a_{s2} = +1$). This BAMDP can be decomposed into several MDPs if knowledge of the transition dynamics is available. For example, action 1 leads to the same state that the agent is in and action 2 to the other state. Another MDP could account for transitioning from state 1 to state 2 using action 1, and from state 2 back to 1, using action 2.

5.4.2 Mathematical Formulation

To illustrate the above, we assume a simple two state MDP, with two actions as in fig. 5.4. In this MDP the transitions are unknown. This means that the agent can move from state 1 or 2 to each of the states 1 or 2 with whichever action $a \in \{1,2\}$, according to the probabilities $p^a_{ss'}$. We will use this type of approach to model a subject's uncertainty on the task dynamics, when encountering this task for the first time.

As described above, we assume that a participant will go through a simulation

phase in which he or she evaluates action trajectories according to a belief over the task dynamics, and later updates this belief according to what experiences they have encountered by interacting with the task. Eventually, the participant will attempt to recover the true MDP underlying the task. Problems with such a characterization can be described by Bayes-Adaptive MDPs (BAMDP) (Duff, 2002). The concept of MDPs are already given in detail in section 2.2.1 but we repeat here for completeness.

Markov Decision Processes (MDPs): An MDP is a model for controlled random processes in which an agent's choice determines the probabilities of transitions of a Markov chain and lead to rewards. Formally, an MDP is a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$ where $\mathcal{S}$ is a finite set of states, $\mathcal{A}$ a finite set of actions, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ a state transition probability matrix with elements $p^a_{ss'} = P(s'|s, a)$, indicating transition from state s to s' by selecting action a, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ a reward function and $\gamma \in [0, 1]$ a discount factor. When all the components of the tuple are known, standard dynamic programming algorithms can be used to obtain the optimal value function.

Bayes-Adaptive Markov Decision Processes (BAMDPs): To explain what a BAMDP (Duff, 2002) is, it is important to introduce a new concept, the *hyperstate*: (s, h). A hyperstate consists of two components, the physical state s of the Markov Chain and the information state h, which summarizes past history of the transitions between the physical states. As we will see, a sufficient statistic to describe a history of past visited states and actions can be given by the counts of how many times a state was visited and how many times an action from that state was executed.

In general, the transitional dynamics are unknown and we thus assume a prior distribution $P(\mathcal{P})$ over all possible models of the environmental dynamics. After observing a history of visited states and actions $h_t = s_1, a_1, s_2, a_2, \ldots, a_{t-1}, s_t$ from controlled interaction with the MDP, the posterior belief about the underlying dynamics of the MDP updates according to Bayes' rule $P(\mathcal{P}|h_t) \propto P(h_t|\mathcal{P})P(\mathcal{P})$. In this way, the agent iteratively attempts to approximate the real dynamics of the MDP, and eventually builds a model of the environment (i.e., transition probabilities and reward function).

We illustrate the above with a simple example described in Guez et al. (2013b).

Let us assume two MDPs (fig. 5.5) represented as trees. Each one has a prior probability of occurrence 0.5 ($\mathcal{P}_0 = \mathcal{P}_1 = 0.5$). The two MDPs are episodic starting from state s_0 and finishing at the leaves of each tree. We consider a case in which an agent has uncertainty over state transitions (i.e., which MDP it is facing) which leads to difficulty in planning. Under the prior distribution over the MDPs, any action from state s_1 to state s_2 has expected reward 0. Thus, in state s_1 the agent cannot identify which one of the two MDPs it is facing in order to take an informative decision. However, the outcome of a transition from state s_0 and action a_0 is critical for the agent to identify the MDP that is in. With this information the agent will have knowledge of the environment it faces with all the transition probabilities and thus can make an informative decision in state s_1.

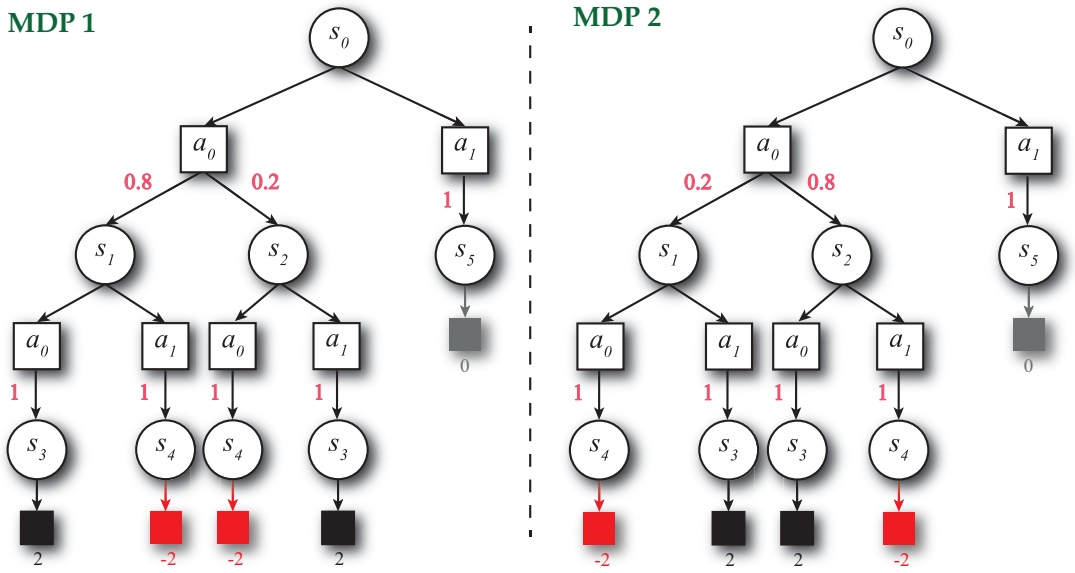


Figure 5.5: A BAMDP decomposed into two MDPs. A BAMDP can be represented as multiple candidate MDPs. In case that an agent is facing a task in which it cannot distinguish which MDP it is facing, in order to take an informative decision at each time step, it needs to maintain and update a belief over possible MDPs (Bayes-adaptivity).

A question that an agent needs to answer at every time step is which tree is the correct one. To tackle this, the agent maintains a belief over the possible trees and updates this belief according to its observations. For all these to take place, the agent maintains a belief over the possible MDPs. According to Bayes rule the probability of being in MDP1 given transition history $h_t = s_0 a_0 s_1$ is $P(\mathcal{P} = \mathcal{P}_0 | h_t) \propto P(h_t | \mathcal{P}_0) P(\mathcal{P}_0) = 0.8$–after normalization. Therefore, the agent knows that if the

resulting state is 80% the state s_1 then the current MDP is the MDP1 ($\mathcal{P}_1$). The same applies if the resulting state is 80% the state s_2 and then the current MDP is MDP2 ($\mathcal{P}_2$). This result demonstrates how critical is the Bayes-adaptivity of the agent in order for it to make informative decisions.

Now, we can proceed to the formal definition of a BAMDP. From the above example, a state can be augmented with the past history to form a new type of state the *hyperstate*. The new state space now consists of a set of all possible histories denoted by $\mathcal{S}^+$, where the symbol "+" is used to indicate the augmentation of the physical state space with the history space. The probability of observing hyperstate (s', h') after executing action a at hyperstate (s, h) can be given by the predictive posterior distribution

$$\mathcal{P}^+((s,h),a,(s',h')) = \int_{\mathcal{P}} \mathcal{P}(s,a,s')P(\mathcal{P}|h)\,\mathrm{d}\mathcal{P} \tag{5.6}$$

and $\mathcal{R}^+((s,h),a) = \mathcal{R}(s,a)$. The tuple $(\mathcal{S}^+, \mathcal{A}, \mathcal{P}^+, \mathcal{R}^+, \gamma)$ forms a Bayes-Adaptive MDP. To solve a BAMDP we will use state-of-the-art algorithms which are based on tree search methods (Monte Carlo Tree Search–MCTS) combined with algorithms that solve bandit problems (UCB1, Auer et al. (2002)), resulting to the UCT algorithm (Kocsis and Szepesvári, 2006). The MCTS tree search algorithm is very efficient in a BAMDP case as the state space of all possible histories can be very large. After practical modifications the combination of all these methods results in the Bayes-Adaptive Monte Carlo Planning algorithm (BAMCP) (for an in-depth treatment refer to Guez et al. (2012, 2013b)).

In our approach, when the participant interacts with the real world, he or she is trying to solve a real MDP, using an approximate model of the dynamics of that MDP (all transitions can be assumed possible). The experience gained in the real world is used to update the approximate model which gradually will resemble more accurately the real world MDP. This updating process can be summed into the following steps: the participant samples an MDP according to the posterior $P(\mathcal{P}|h_t)$, solves it by using planning and thus selecting an action which is considered most rewarding, applies that action to the real environment, and finally updates the posterior belief over the dynamics of the task according to the new observations received (the feedback from the environment).

As we will argue, the choice of the prior distribution describing the transitional dynamics, which can be extended to uncertainty about rewards, is crucial for the

performance of the agent. We assume that infants have a less precise prior than adults, which should result in less optimal behavior in goal-directed tasks. On the other hand, having an imprecise prior might also provide flexibility in learning and adapting in different tasks, whereas the structured prior of an adult might bias his simulation phases and result in a less optimal solution and eventual model.

In the corresponding literature (Dearden et al., 1999; Strens, 2000) the simplest prior that can be used is a Dirichlet prior over the transitional dynamics. This prior has very good analytic properties in conjunction with a Multinomial likelihood. The Multinomial likelihood can successfully describe the probability of the occurrence of one-of-K discrete outcomes. According to this we assume that

$$p^a_{ss'} \sim Dir(\alpha) \tag{5.7}$$

$$s' \sim Mult(p^a_{ss'}) \tag{5.8}$$

This form of the prior results in an analytically computed Dirichlet posterior distribution. To compute the predictive posterior as in eq. 5.6, we calculate the expected value of the transitional probabilities, over all possible models $\mathcal{P}$, weighted by their posterior distribution. To update the posterior distribution, according to Bayes' rule, we need only to add to the α parameter of the Dirichlet distribution the counts of observing a particular event (in our case states occurred in history). The counts of visits are sufficient statistics for a random variable multinomially distributed. In our case states are generated by a multinomial distribution, thus the history of states visited can be described by the visit counts.

Summing up, we described two processes: In the planning phase our virtual agent tries to act optimally in a completely or partially unknown environment (BAMDP). It simulates possible future action sequences and estimates their value. According to this, in a subsequent real-world interaction phase, the agent selects an action in order to achieve an optimal future outcome. Then, according to the following observation and reward the agent receives, it updates its approximate model of the environmental dynamics. This procedure repeats until a policy for the real underlying MDP has been learned. In our formalization of these processes, we adopt the BAUCT algorithm developed by Guez (2015).

5.4.3 Experimental benchmark task: The Toybox

The toybox of Monroy et al. (2015a; 2015b) offers a good testing environment in order to investigate the plausibility and performance of the above framework. As described above (see section 5.2.1), the experimental procedure consisted of a demonstration video, during which participants observed an action sequence comprised of underlying transitional probabilities between all possible actions, which were initially unknown to the participants but which could be learned after repeated observations. After this initial phase, subjects were introduced to the real toy stimulus and could then freely select their own actions. This action sequence further included a causal event- the distal action-effect- which could also be learned by the participants and then achieved by themselves.

A straightforward initial approach for modeling the participants' learning processes, which will be used as prior information for the task dynamics is: the observer is introduced to an environment via the demonstration video, during which we assume that the participants attempt to learn a model of how the toy-box 'works'. Additionally, we assumed that humans tend to perceive other human actions as goal-directed, and are biased to perceive effects as having a cause. Here, we define a state as fixating on an area of interest (AOI) on the presentation screen, as this represents the physical location of where observers perceive each action when it occurs. A model-free approach for learning the transition matrix (i.e., the set of transitional probabilities between all possible AOIs) could be followed: For a transition from state s (fixating on a specific AOI) to s' (fixating on the subsequent AOI) we update $T(s,s') = T(s,s') + \eta(1 - T(s,s'))$. For all other states s'' other than s', the probabilities are reduced according to $T(s,s'') = T(s,s'')(1-\eta)$, to ensure that the whole distribution remains normalized.

First, this approximate model of the toy-box transitional dynamics can be used to test the hypothesis that an initial demonstration phase does indeed inform goal-directed behavior during the second phase, in which the participant interacts with the toy-box. Second, a hypothesis can also be made that the action effect (the light turning on) provides a reinforcer that increases the observer's ability to learn the correct causal sequence preceding the light's occurrence. We adopted a model-free way for modeling the learning of transitional probabilities from AOI to AOI.

In the second phase, a model (BAMCP) that will combine the learned domain structure from the demonstration phase, could be tested. The model-based/planning section of the algorithm will use the approximate model that was possibly learnt

from the demonstration procedure (based on eye-tracking data). This model can be learned in a model free way like in Gläscher et al. (2010). Some transition matrices from experimental data are given in fig. 5.6, 5.7, 5.8. These figures represent the probability of an eye-fixation transition from an AOI to another AOI. More specifically, we calculated the frequency of which a participant shifts his/her eye fixation from one AOI to another AOI.

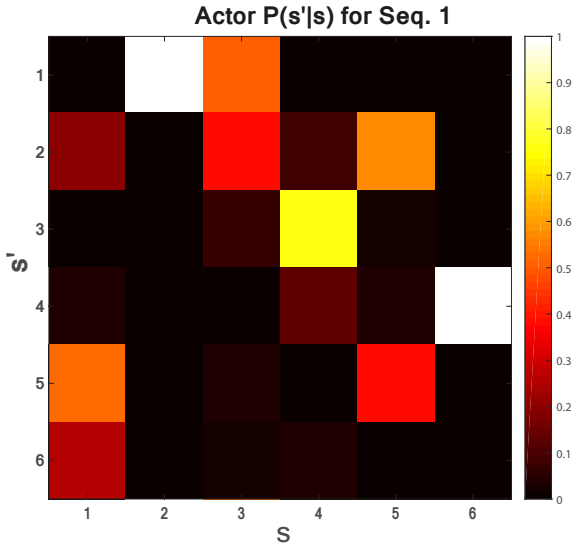


Figure 5.6: Transition matrix from the actor's movements. In this case, the probabilities of transitions are the observed empirical frequencies of the transitions of actor's movements from a state s' to a state s. For the particular sequence of moves examined (Seq. 1), the AOIs that correspond to the Effect pair are 2 and 1, for Action 1 and Action 2 respectively. The AOIs that correspond to the Non-effect pair are 6 and 4, for Action 1 and Action 2 respectively. The AOI 7 corresponds to the light. White color indicates high probability of transitioning at this AOI. The transition probabilities of the Effect and Non-effect pairs appear with probability one. These transitions are the transitions that we assume the participants might try to learn at the demonstration phase.

5.4.4 Bayes-Adaptive Planning in Toybox

The model, described in the previous section, that learns the transition probabilities of a participant's eye-fixations, assumes that the underlying toy-box real transitional dynamics can be approximated in a model-free way. During the real interaction with the stimulus, the participant's model of transitional dynamics is not

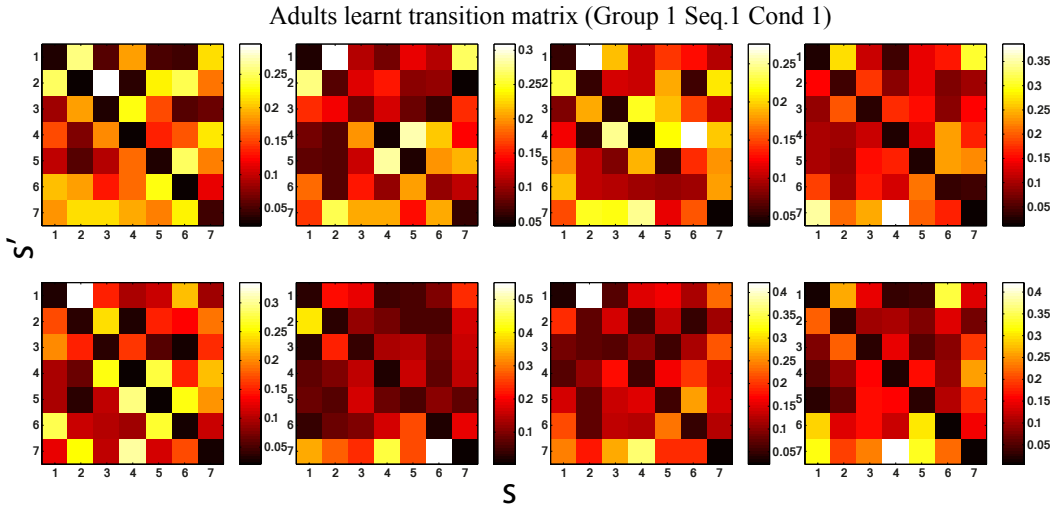


Figure 5.7: Adult matrices of transitions of their eye-fixations. Transition matrices learned using a model-free algorithm of 8 adult participants. The AOIs that correspond to the Effect pair are 2 and 1, for Action 1 and Action 2 respectively. The AOIs that correspond to the Non-effect pair are 6 and 4 for Action 1 and Action 2 respectively. The AOI 7 corresponds to the light. White color indicates high probability of transitioning to this AOI.

updated, according to the sampled experience acquired during the participant's interaction with the toy-box but rather all possible transitions are considered equally valid and optimal.

According to the BAMDP modeling framework discussed in the previous section, an alternative approach would be to consider the whole experimental procedure as a unified process in which during the observation phase the participant constructs a prior over the possible transitional dynamics of the MDP (which are initially hidden), and during the execution phase this prior is used to approximate the underlying MDP. One way to construct the real underlying MDP is to consider three states: 'light off', the latent state (i.e., one step before switching on the light), and 'light on'[4] (fig. 5.9).

We used the eye tracking data from the actual experiment to create a prior $P(\mathcal{P})$

[4]Alternatively we can consider the states of the toybox in relation to the light state. For example, the toybox could have two states: state 1, where nothing happens and state 2 which is the state of the toybox after performing the first action of the effect pair. Performing the second action of the effect pair the light is turned on and the toybox resets to its first state.

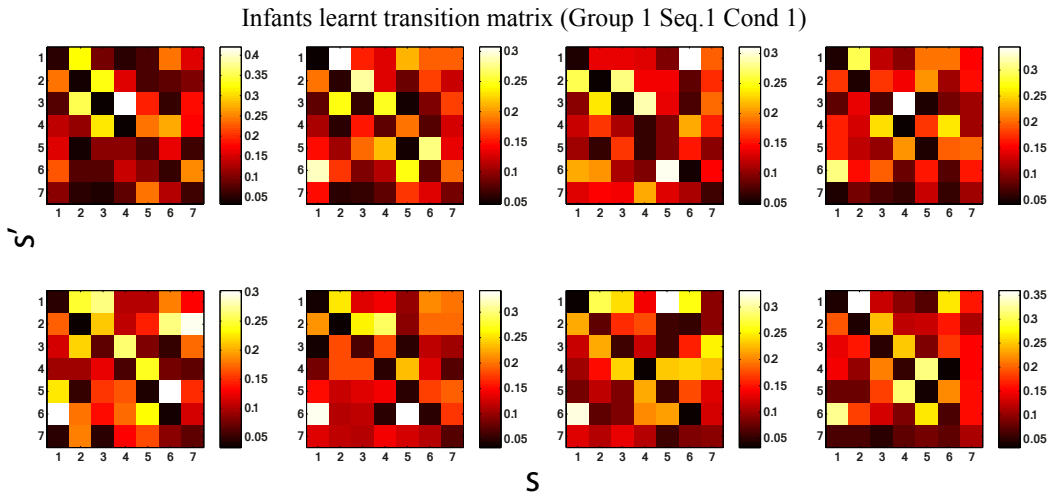


Figure 5.8: Infants matrices of transitions of their eye-fixations. Transition matrices learned using a model-free algorithm of 8 infant subjects. The AOIs that correspond to the Effect pair are 2 and 1, for Action 1 and Action 2 respectively. The AOIs that correspond to the Non-effect pair are 6 and 4 for Action 1 and Action 2 respectively. The AOI 7 corresponds to the light. White color indicates high probability of transitioning at this AOI.

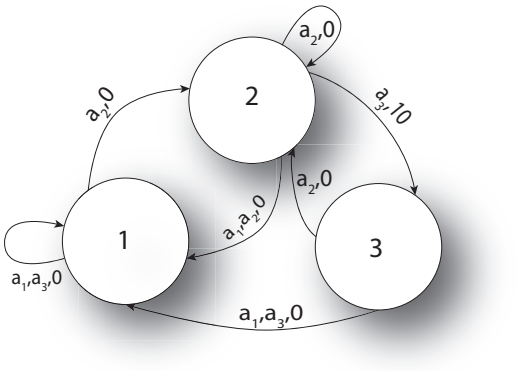


Figure 5.9: The real MDP underlying the toy-box. Assuming that a_2 is the action that transits the system to the latent state (i.e., one step before switching on the light), a_3 the action that leads from latent state to a 'light on' state and a_1 the rest of the actions. Switching on the light rewards the agent with $r = 10$ and all other actions return $r = 0$. From the 'light on' state the toy-box can get back to the state 2 by using action a_2 or to state 1 by using all other actions a_1.

over transitional dynamics for each individual participant. First, we implemented the model-free approach (discussed in the previous section) to update the transition matrix, and we used the resulting matrix as a prior over the possible transitions

during the action execution phase along with the BAUCT algorithm to model each participant's behavior. The reason we used the eye-tracking data is based on the patterns we found in these data, that reveal that participants can infer the actions that cause the light to switch on. This can be interpreted as a sign of participants' understanding (fully or approximately) of the underlying dynamics.

Our intention is to fit the model described by the BAUCT algorithm to the behavioral data, using the prior knowledge demonstrated in the eye-tracking data, acquired at the demonstration phase. With such models, it is very difficult to compute likelihoods of the form $P(\mathcal{D}|\mathcal{M})$, where $\mathcal{D}$ is the data and $\mathcal{M}$ is the model, thus we have to rely on likelihood-free methods. In brief, we iteratively simulate data $\mathcal{D}$ from a prior distribution $P(\mathcal{M},\theta)$ over possible models $\mathcal{M}$ and model parameters θ, and we compute features $\phi(D)$ (i.e., summary statistics). We then compare the model's summary statistics with the observed summary statistics from participants' action sequences and we reject or accept this sample if the difference is smaller than a criterion acceptance rate. The whole process is described in detail in Chapter 3.

In this experiment, features are not easy to extract, given that the behavioral data were too few, and that participants have no prior instructions on what to do with the toy-box (we assumed that they tried to switch on the light by learning the proper transitions, although this might not be entirely true). We ran simulations with the BAUCT algorithm and observed the action sequences generated by the algorithm. We noticed that the BAUCT after many iterations learns how to switch on the light. It remains to be examined how this model fits to the human data with the Approximate Bayesian Computation, as described in Chapter 3.

5.5 Associative Approaches

Associative learning models (Courville et al., 2006; Le Pelley, 2004; Mackintosh, 1975; Rescorla et al., 1972) specify rules for the development of predictions given the stimuli presented. These models will be discussed in detail in section 5.6.3.1. Each stimulus is assumed to have an associative strength, which characterizes how strongly it predicts reinforcement. In our case, according to the results of Monroy et al. (2015a; 2015b), participants were able to correctly predict the light effect (reinforcer). Thus, we assumed that each stimulus should have a weight characterizing the prediction of the light effect. These associations should reveal higher

values when the actor is performing an action from the effect pair which preceded the actual occurrence of the effect. The weights are updated recursively and proportionally to the reward received.

It has been shown that humans respond to surprising events (i.e., when there is uncertainty about them) with faster learning (Courville et al., 2006; Dayan and Jyu, 2003). To capture this behavior within our computational model, an attribute known as 'associability' was added to the recursive updating of the associative strengths. This attribute tunes how quickly a particular associative strength is updated. For example, the Pearce-Hall model (Pearce and Hall, 1980) updates the associative strength V_i for each stimulus i present at time t according to:

$$\Delta V_i(t) \propto \alpha_i(t)\lambda(t) \tag{5.9}$$

where $\lambda(t)$ is the magnitude of the reward delivered and $\alpha_i(t)$ is the associability of the stimulus. The associability is further modulated by surprise as:

$$\alpha_i(t) = |\lambda(t-1) - V_i(t-1)| \tag{5.10}$$

which is the absolute value of the difference between the actual reward λ and the reward predicted V_i by the model from the preceding trial.

In our case, by carefully observing fig. 5.10, we notice that for each actor's action there are several fixations made by the participant to various AOIs. Thus, we assume that the greater the number of fixations to a particular AOI, the greater also should be the value of the total duration of these fixations. We expect these durations to have a higher value just before an event actually happens (i.e., the light), if the participant is predicting an outcome from that AOI (and thus is fixating his gaze towards the predicted location). Intuitively, we attempted to use a similar model as in Pearce and Hall (1980) to explore possible patterns of people's gaze fixations. These patterns might reveal the signature of the learning process that takes place.

5.5.1 Rationale I

In this section we present a second attempt to model the learning process that takes place during the toy-box task. We assume that learning takes place during the demonstration phase, and is then available to exploit during the action execution

phase. Thus, we reasoned that the eye-tracking data can be used as input into a learning model. This reflects the idea that there should exist a process by which the participant associates the actor's actions with the effect in a causal way.

Our initial approach was to use the eye scan paths (i.e., each consecutive AOI visited) as a temporal set of participants' choices and fit a model-free model, inspired by Hayes et al. (2011). We tested how a model-free model fits the data, and attempted to infer the learning rate parameter α. However, the inferred parameter resulted in odd values (very close to zero). As we discussed in Chapter 3, model fitting might lead to odd values of the inferred parameters for various reasons. In the approach we followed in this section, there are a lot of factors that might have led to these results.

One issue with the approach followed here is that it was not possible to directly associate one of the actor's action with the gaze behavior of the participant. This is because the gaze scan paths are at a different temporal scale than that of the actor's actions, so it is possible to make many distinct gaze fixations during the time window that corresponds to one of the observed actions. Furthermore, there may be repeated transitions that do not reflect learning processes but are rather a feature of gaze behavior, which is inherently noisy. For example, a person might make several fixations in different AOIs but their duration is small, or they might make several fixations that fall within an AOI as they move their eyes from one location to another, which does not necessarily correspond to their underlying expectations or beliefs. In our analysis, we did not weight fixations according to their durations, but instead we created scanpaths as we were interested in eye transitions among AOIs. This might bias the RL model's value function towards fixations that are not informative of the participant's real preferences (e.g., involuntary eye movements, information gathering, distractions, etc.).

5.5.2 Method I

To associate the participant's gaze fixations with the actor's actions, we transformed all of the gaze fixations that occurred within each time interval (one time interval corresponded to one observed action) into a probability distribution (fig. 5.11). This distribution can be seen as a reflection of the participant's beliefs about the underlying transition matrix during the observed trial (i.e., action). Furthermore, although we expected the fixations to be biased towards the actor's action at the current time interval, we assumed that any fixation to another AOI would reflect

	1	2	3	4	5	6	7
Actor	Push	Squeeze	Slide	Open	Spin	Bend	Star
5	0	0	0	1	2	0	0
3	0	0	3	1	0	0	2
3	0	0	5	1	0	0	1
2	0	0	1	0	1	1	0
1	3	1	0	0	0	0	2
1	2	1	2	0	0	0	2
4	0	0	0	2	0	0	2
2	0	0	1	0	0	0	3
1	1	0	0	0	0	0	1
4	0	2	1	2	0	0	0
1	2	0	0	0	0	0	3

Figure 5.10: Actor's actions and subject's fixations. Actor's actions are coded with real integers ranging 1-6 as well as a subject's eye fixations ranging from 1-7. Actor's actions are presented in the first column of the table whereas AOI's are represented by the following 7 columns. Within the actor's action time interval, the subject makes a number of fixations at different AOIs. When the first action of the Effect pair is executed by the actor (coded as 2 and highlighted with yellow color), the subject expects the star-shaped light to turn on. Thus, more fixations are observed in the 'star' AOI. During the second action of the Effect pair (coded as 1 and highlighted with dark yellow) subjects made even more fixations to the light's AOI. It seems that the first action of the Effect pair is perceived as a predictor of a predictor (in our case, the action preceding the light effect is a predictor of the light effect).

a prediction about what action was most likely to occur next. This was evident according to the results of Monroy et al. (2015a; 2015b).

As a first attempt to capture the pattern of fixations, we assumed that the number of fixations at a specific AOI is following a Multinomial distribution. Using the conjugate distribution of the Multinomial distribution, as a prior we applied Bayesian online updating, according to the subject's current fixations, of this simple Dirichlet-Multinomial model. More specifically we assumed that a fixation at a particular AOI is represented as a 1-of-7 random vector following a Multinomial distribution, $\mathbf{x} \sim Mult(\mathbf{p})$, where $\mathbf{p}$ are the parameters of the distribution. A prior distribution over the parameters is assigned to a Dirichlet distribution $\mathbf{p} \sim Dir(\boldsymbol{\alpha})$ where all α were set to $1/7$. The posterior distribution of the probability of fixating at a particular AOI, is also Dirichlet, $\mathbf{p}|\mathbf{x} \sim Dir(\boldsymbol{\alpha})$, because of the conjugacy of the two distributions. Starting with a prior distribution, the online update consists of updating the pseudocounts $\boldsymbol{\alpha}$ of the Dirichlet, by the number of fixations counted during each time interval (a time interval corresponds to one action). Thus, if we

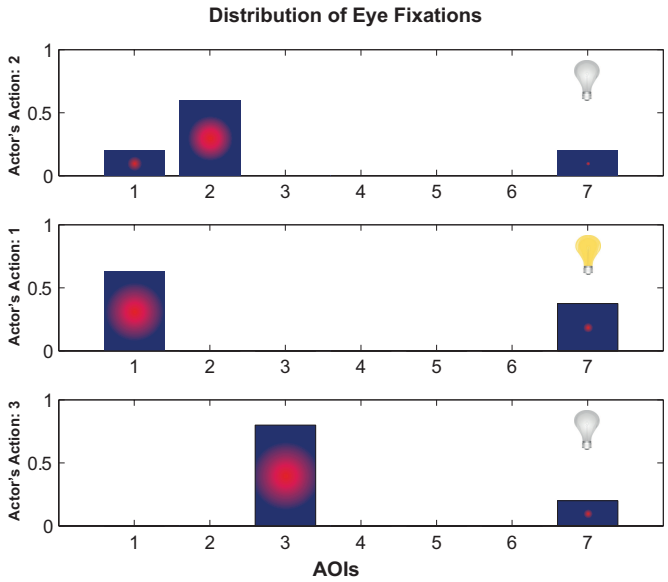


Figure 5.11: The distribution of eye fixations of one participant within 3 time intervals (equivalent to 3 actor's actions). The radial gradient inside the bars, indicates the percentage of time spent at the corresponding AOI, during actor's action time interval. For example, small radius of the gradient represents that the subject spent, in total for that time interval, less time fixating that AOI. At this time interval, the actor is executing an action on the toy-box which results in a 'light-off' or 'light-on state', denoted by the light bulb color. The participant has already been exposed to 25/96 actor's actions and the shift in attention towards the AOI of light is apparent. Furthermore, the participant being familiar with the actor's action transitions, he or she can predict when the light is about to turn on, thus there is an amount of fixations also at the AOI of the light, which increases when the actor is performing the second action of the effect pair.

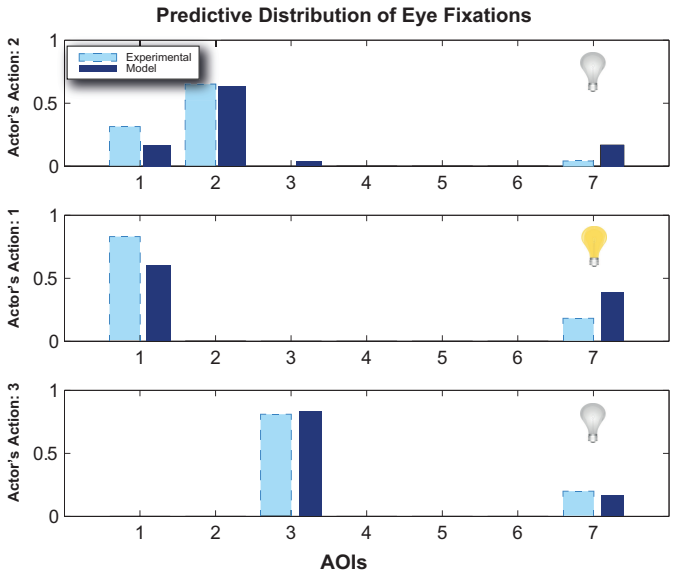


Figure 5.12: Predictive Distribution of eye-fixations for a participant at the same time intervals as in fig. 5.11 (action coding scheme remains the same (i.e., effect pair: 2,1)). The online Bayesian model predicts the probability of fixating at a specific AOI according to the observed number of occurrences of particular transitions among actor's actions.

start with $Dir(\alpha)$ and at the next time interval we observed $n_1,\ldots,n_7$ counts for each one of the 7 AOIs, the posterior will be $Dir(\alpha+n_1,\ldots,\alpha+n_7)$. Given some new observations $\mathbf{n}$, and having observed data $\mathbf{X}$, the probability of fixating at the AOI_j is given by the posterior predictive $P(\mathbf{x}=j|\mathbf{X})=\frac{\alpha_j+n_j}{\sum_{j'}(\alpha_{j'}+n_{j'})}$.

5.5.3 Results I

The results of this modeling procedure revealed that the online Bayesian learner managed to predict the probabilities of fixating at any AOI during the next time step (fig 5.12). It seems that participant's predictions are more accurate when the actor's action is one from the Effect pair. This might indicate that the AOI preference has to do with how much the participant values the particular AOI according to a cause/effect. This value might be a measure of interest or expecting surprise for that AOI. It remains to link this value of each AOI to the time spent on fixating at this AOI at a time interval. Bringing these two descriptions under a common framework might also shape a theoretical framework for attention during decision making processes.

5.5.4 Rationale II

Longer time spent looking at an AOI, usually indicates that the observer expects something interesting should happen in that location. Under this assumption, we implemented an adaptation of a RL model to track fixation duration at each of the seven AOIs (6 "buttons" plus the light area of interest). Instead of estimating expected reward, the model estimates the expected duration of fixation at a specific AOI. The choice of "where to look" is made by using the value function of the duration within a particular AOI, inserted in the Boltzmann function (eq. 2.16), which was used in previous chapters. In general, the probability distribution over fixations given by the model closely matches the proportions of fixating at a specific AOI at each time interval from participants' data as can be seen in fig. 5.14.

5.5.5 Method II

To associate AOIs with the actor's actions, in order to strengthen their relationships according to subjects perspective, we used the simple Rescorla-Wagner (RW) rule

$$Q_t(a, AOI) = Q_{t-1}(a, AOI) + \alpha(dur(t) - Q_{t-1}(a, AOI)) \tag{5.11}$$

where a is the actor's action at time interval t, α is the learning rate and the reward prediction error is the difference between actual fixation duration $dur(t)$ at time-step t and the prediction of the model from the previous time-step $Q_{t-1}(a, AOI)$. For selecting the location of fixations, the model computes a probability distribution, using the Boltzmann function (eq. 5.15), by taking into account the expected duration of fixation Q during an actor's action. It is worth noting that by comparing the proportion of fixations at each AOI per time interval the Q-learning model with the Boltzmann distribution for AOI selection method shows reasonable results (fig. 5.13). However, a Bayesian learner that learns only transitions (a Dirichlet-Multinomial model) learns the probabilities of fixating at a specific AOI better.

5.5.6 Results II

Fitting the model (eq. 5.11) to the actual data, we can infer the parameters of the model (learning rate α, exploration-exploitation trade-off β). By maximum likelihood, the infants mean learning rate was $\alpha_{inf} = 0.27$ and for adults $\alpha_{adul} = 0.34$

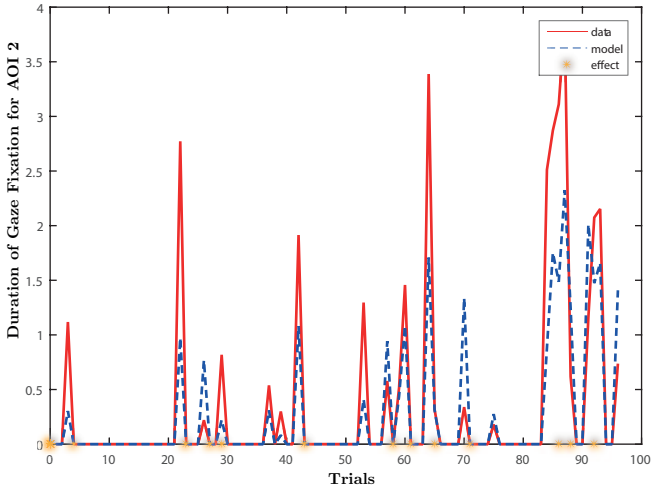


Figure 5.13: Model Predictions for the AOI 2. The AOI 2 corresponds to the second action of the Effect pair. The y-axis is the Q function which represents the expected duration of an eye-fixation at the AOI 2. The x-axis represents the time frame in which the actor acts. It is split in 96 intervals in which each one of them the actor performs an action. We notice a high expected duration value just before the light turns on consistent with experimental data.

(significant difference $p = 0.005$ under a t-test). Thus, the model confirms that both children and adults learn during the demonstration, but adults learn more quickly. With these inferred parameters we can simulate the whole participant's behavior (i.e., using the model and the inferred parameters to simulate actions). In fig 5.13 is presented an example of the model's behavior compared to the subject's actual duration of fixation patterns within a specific AOI (this AOI is the second action of the effect pair). In addition, we also present the probability of fixating at the first AOI of the effect pair as it evolves in time (fig. 5.14).

5.5.7 Discussion

The value function can be interpreted as a measure of expectation or in general the value of this particular AOI according to the subject. According to our model, the value function tracks the expected duration of fixation, while the Boltzmann distribution uses the value function to estimate the probability of fixating at a specific area. With the framework we described, the subject's data are used to extract the patterns given by the fig. 5.11. Extensions and further research could include a varying step-by-step learning rate such as in a Kalman filter, which could reveal differences in trial-by-trial learning rates of adults and infants. Another

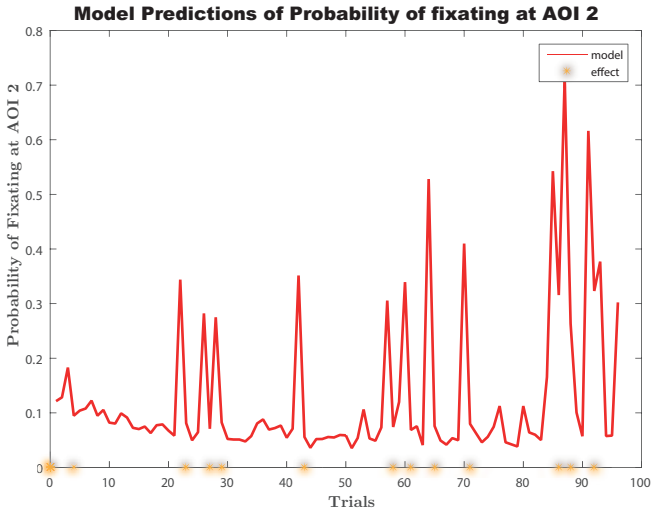


Figure 5.14: The probability of fixating at a specific AOI given the time intervals where the actor acts. The probability increases as the subject learns to expect the light. Similar patterns for other AOIs weren't observed.

potential direction would be to implement a polynomial or exponential function for the learning rate, and potentially examine the form of this function and the differences among age groups.

5.6 Model Free Learning Rules and Conditioning

5.6.1 Introduction

Learning from observations is an important source of information for humans, and in particular for young infants and children who have fewer experiences that they can rely upon when selecting their own actions than adults do. In the previous section, we described model-free reinforcement learning techniques that were employed to investigate the learning of an observed action sequence in both infant and adult populations. Eye-tracking data suggested that participants were able to detect deterministic action pairs embedded within a sequence of six possible actions, and demonstrated the ability to make correct gaze fixations to the location of subsequent actions in the sequence before they occurred.

Crucially, the data from Monroy et al. (2015a; 2015b) indicated a possible link between both infant and adults' predictive gaze behavior and their action execution following the learning observation phase. We formulated a model in an

attempt to capture the underlying learning mechanisms that gave rise to this predictive behavior. The following analysis attempts to give a normative explanation of how subjects learn from observing another agent's actions, and then transfer this knowledge to the action execution phase.

During the video, participants were exposed to the actor's action, following which (or during which) they make a decision on where to look. Without being given any prior instruction, participants tended to be better at predicting the location of the next action when it preceded a salient effect (i.e., the light), or at predicting the location of the light itself. We can consider each observed action as a stimulus, from the subject's perspective, which is somehow related to the other actions and eventually the light effect. Thus, participants should associate all possible actions with each other and with the light effect to a certain extent. This association will become strengthened or weakened depending on where the participant looks and whether or not an event occurs at that location.

For each observed trial (a trial refers to one action in the observed sequence) we determined the particular AOI towards which a participant made the maximum duration of fixation time. For this measure, we summed the total durations of all individual gaze fixations for each AOI. This assumption is based on evidence from prior research on the relationship between attention and eye gaze fixations (Aslin, 2007). Aslin and colleagues showed that, in a visual attention paradigm, increased total looking duration reflects increased attention towards the associated AOI. It is generally accepted that looking times might reflect a combination of a) stimulus-driven attention, (b) memory of past stimuli, and (c) comparison between the current and the past stimuli (Kidd et al., 2012).

5.6.2 General Method

To account for the associations between an observed action and the AOI towards which the participant looked the longest, we implemented a system of rewards as follows: if a participant predicted correctly the next action, we assigned a reward $r = 1$, otherwise the reward given was $r = 0$. Here, we emphasize that we did not distribute a reward only when the participant predicted the light effect. Instead, we allowed the model to receive rewards for any instance in which the participant made a correct prediction. This allowed us to take into account the entire structure of the experiment, in which Non-effect and Effect pairs of actions were demonstrated with the same frequency and thus provided equal opportunities for learning and

prediction.

As described above, our data were transformed into a time-series of choices (i.e., gaze fixations) for each AOI to which the participant attended to or made a predictive fixation, and we assigned rewards according to when these these fixations were correct or incorrect. These choice-reward pairs can then serve as inputs to a RL model, that will attempt to describe the computations that the learner makes during the observation phase. However, we studied the learning phase in the observation phase isolated from the behavioral phase. Next, we aimed to correlate the gaze behaviors during the observation phase with the action execution data from the second phase of the experiment.

5.6.3 Three Models

This subsection presents three nested models of the acquisition of looking preferences. The first uses Rescorla-Wagner learning rule with a simple Boltzmann function for selecting an AOI to look. The second extends this by adding Temporal Difference learning. The final model retains the Temporal Difference rule of the second model but elaborates the action selection function by including an attentional component.

5.6.3.1 Rescorla-Wagner Rule

We describe here the main modeling framework which all associative models that we explored are dependent upon. For this, we employed the well-studied Rescorla-Wagner learning rule (Rescorla et al., 1972). The model learns to assign an action value $V(c)$ to each choice c according to previously experienced rewards. These functions are learned by a delta rule: if choice c was chosen and reward $r \in \{0,1\}$ received, then $V(c)$ is updated according to:

$$V_t(c) = V_{t-1}(c) + \alpha \cdot \delta_t(c) \tag{5.12}$$

$$\delta_t(c) = r_t - V_{t-1}(c) \tag{5.13}$$

where the α parameter controls the learning rate. Note, the learning rate is considered fixed throughout the entire observation phase. This learning rate controls the extent to which the prediction error $\delta(c)$ affects the estimation $V(c)$. Specifically,

when a participant perceives an action, he produces gaze fixations to several possible AOIs with varying durations. We assume that the participant is influenced by observing certain subsequences that result in a distal effect. From the series of various fixations, we assumed that the one with the highest duration is the one with a particular interest for the participant. For this choice of AOI, the corresponding value function is updated according to eq. 5.12 and eq 5.13.

According to the RL framework, a participant makes a prediction and fixates to an AOI, then updates the corresponding value function according to the previous value function estimation and the amount of reward prediction error after observing the next action, modulated by the learning rate. We expect that an ideal learner takes into account more the prediction error rather than relying on his own estimations, and thus adjusts his predictions better during the whole task. This indicates a higher learning rate (fast learner) whereas a participant that relies on his own estimations and regards less the prediction error, which represents the error between reality and estimation, ultimately will reveal a lower learning rate (slow learner).

As we mentioned above, participants were not told about the causal relationship (i.e., the Effect pair of actions that caused the light to turn on) and had to infer it based on their observations. This could also be considered as a form of second-order conditioning, which cannot be captured by a simple RL model as that described above. Second-order conditioning is defined as when a stimulus CS1 is followed by an unconditioned stimulus US and then another stimulus CS2 is paired with the first one, predicting the upcoming of the US. In our case, the second action of the effect pair (CS1) could be first associated with the light effect (US) and then the first action of the effect pair (CS2) is associated with the second action of the pair.

It is worth repeating that Action 1 is always followed by Action 2 during the whole course of the observed sequence. Of course, a participant might be conditioned into more actions prior to the appearance of the effect pair which accounts for higher order conditioning. For example, during the interaction phase some participants performed another action before performing the effect pair thinking that that particular action also belonged to the set of actions that turned on the light.

5.6.3.2 Temporal Difference Rule

Here, we consider the possibility that a model that can account for second-order conditioning could better explain the observed experimental data. For this, we

used the Temporal Difference (TD) learning rule which uses eq. 5.12 to update the value functions but introduces a different form of reward prediction error:

$$\delta_t(c) = r_t + V_{t-1}(c') - V_{t-1}(c) \tag{5.14}$$

where c' refers to the learner's future choice that would select one time step ahead according to his current policy. It is worth mentioning again that the value function is the expected reward from that particular choice. Eq. 5.14 differs from eq. 5.12 only by the term $V(c')$. This reflects the desire to learn not only the immediate reward but all future rewards following the subsequent choice c'. In our case, ideally, the choice c would be Action 1 and c' would be Action 2 of the Effect pair. In this way, the participant reveals that he or she is conditioned not only by the Action 2, but also by Action 1, as the latter predicts a predictor of a reward.

However, what does this extra term imply for our experiment, and how do we expect it to improve the outcome of our model with respect to explaining the data? During learning, the observer maintains a set of expected reward predictions for every AOI, reflecting how much reward is expected from making a fixation toward a particular choice (AOI).

The aforementioned reward is given when the participant successfully predicts the actor's following action or the light effect. If the light effect does indeed serve as a reinforcer for increasing predictions towards it (i.e., looking at the corresponding AOI when actor is performing Action 2) then the participant should raise the value of the effect AOI during the trial preceding it. The learner, having observed Action 1 (Effect pair) followed by Action 2 and then the effect, uses the value associated with the latter action as a proxy for the following rewards. In other words, he updates his prediction according to the expected rewards that not only immediately follow the current choice of where to fixate his gaze but also from all future ones. This recursive process allows for second-order (or even higher-order) conditioning to occur between the two action events that activate the effect.

5.6.3.3 Retrospective Gaze Behavior

Given the value estimates during a particular trial, participants are assumed to choose between their options $c \in \mathcal{C}$, where $\mathcal{C}$ represents the total number of possible choices, in a stochastic way (adding some noise to the action selection) with

probabilities given by the Boltzmann function:

$$p_t(c) = \frac{\exp(\beta \cdot V_t(c))}{\sum_{i=1}^{\mathcal{C}} \exp(\beta \cdot V_t(c_i))}, \; c \in \mathcal{C} \tag{5.15}$$

where the β parameter controls the exclusivity with which choices are focused on the highest valued option. In other words, when the value of β is high the option with the highest value is very likely to be selected. When β is close to 0, the actions are selected randomly.

Until now, we took into account a 'predictive' general behavior of the participant. However, the participant in many cases does not predict, but instead follows the actor's current movements retroactively. To account for this, we modified the usual 'sticky' parameter that is used in similar behavioral analyses to capture the effect of persisting on the same choice as the previous trial. Modifying eq. 5.15 by adding an extra parameter ϕ, the model can capture the participant's tendency to follow what currently is happening on the screen. We will call it the *bias parameter*, as it reflects the tendency for participants to bias their attention towards the actor's movements. The modified action selection equation will then be:

$$p_t(c) = \frac{\exp(\beta \cdot V_t(c) + \phi \cdot I(c,a))}{\sum_{i=1}^{\mathcal{C}} \exp(\beta \cdot V_t(c_i) + \phi \cdot I(c_i,a))}, \; c \in \mathcal{C} \tag{5.16}$$

where $I(c,a)$ is equal to 1 if the participant looks longer to the AOI where the action is occurring and 0 otherwise. Positive values of ϕ reflect the participant's choice to follow the actor's movements and negative values represent complete avoidance of attention on what the actor is doing.

5.6.4 Model Fitting

We fit the models with maximum a posteriori estimation (using the Optimization toolbox of MATLAB):

$$\hat{\boldsymbol{\theta}}^{MAP} = \arg\max_{\boldsymbol{\theta}} P(\mathcal{D}|\boldsymbol{\theta}) P(\boldsymbol{\theta}) \tag{5.17}$$

where $\mathcal{D} = \{c_t, r_t, t = 1, \ldots, T\}$ the data, consisted of observed choices and rewards. A prior on parameters $\boldsymbol{\theta} = \{\alpha, \beta, \phi\}$ can be decomposed to:

$$\alpha \sim Beta(1.2, 1.2) \tag{5.18}$$

$$\beta \sim Gamma(1.2, 1.2) \tag{5.19}$$

$$\phi \sim \mathcal{N}(3, 2) \tag{5.20}$$

The parameters of the priors are similar to Christakou et al. (2013). For each participant and for each model, we computed a set of parameters $\hat{\boldsymbol{\theta}} = \{\hat{\alpha}, \hat{\beta}, \hat{\phi}\}$ using gradient search (fmincon–constrained optimization MATLAB function), with different starting points decreasing the chance of local optima, over the likelihood of the participant's choices conditioned on previous rewards and choices for each trial. Specifically, we estimated the optimal parameters at the minimum of the log likelihood which is given by the sum of the log-values of the probabilities computed in eq. 5.16) as:

$$\mathcal{NLL} = \sum_t \log p(c_t | V_t(c_t)) \tag{5.21}$$

where the value function is learned according to the model learning rule (eq. 5.12) along with eq. 5.13 and eq. 5.14).

To test whether the models provided a reliable account of participants data, we performed the following analyses. First, the relative degree of improvement of each model, over the chance model (i.e., a model without parameters which selects choices under $p_t(c) = 1/|\mathcal{C}|$where $|C|$ is the set of admissible choices), provides a descriptive index called pseudo-R^2 (see section 4.4.3). This index is defined as $(R - L)/R$, where R is the log-likelihood of the random model and L the log likelihood of one of the models described at the Methods section. Most of the participants' model-fits give an index significantly higher than zero, although due to noise in the data there are participants with low indices.

The baseline model (RW) served as the basic model and all others were derived from it by adding an extra parameter (i.e., nested models). This enabled us to compare the baseline model with the rest of the models, and show how increasing the complexity of the basic model leads to better description of the data using the Bayesian Model Selection (BMS) method of Stephan et al. (2009). In order to do this, we computed log-model-evidences for each participant and for each model. Because the computation of the evidence involves an intractable integral over parameters (BMS is not dependent on the model parameters as they are integrated out) we used the Laplace approximation (Kass and Raftery, 1995), which

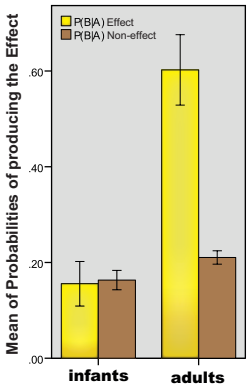


Figure 5.15: Probabilities of each age group for producing the effect pair P(B|A) (Effect) and the non effect pair P(B|A) (Non-Effect). Probabilities were calculated by the empirical frequencies observed in the data, using the conditional probability definition $P(A|B) = P(A,B)/P(B)$.

assumes that the posterior over parameters behaves as a Gaussian around its mode. We then submitted these to the spm_BMS routine from SPM12 (Rosa, 2012) for model selection.

This type of model selection is a Variational Bayes method that treats the model as a random variable, and allows one to compute how likely it is that a specific model generated the data of a randomly chosen subject, as well as the probability of one model being more likely than any other model. Although the usual treatment for model selection is to use Bayes factors (Kass and Raftery, 1995), using the BMS enables us to to use the model evidence for group level analyses without any constraints on the models compared. A great advantage of this method is that the models do not necessarily bear a hierarchical relationship to one another (i.e., they do not need to be nested)(Stephan et al., 2009).

5.6.5 Behavioral Results

To link eye tracking data with behavioral data, further analyses were conducted. Some adults, and very few infants, reproduced the observed Effect and Non-effect pairs when acting themselves. However, adults were more likely to produce the Effect pair rather the non effect pair, as shown in fig. 5.15.

The original data-set consisted of 44 adults and 66 infant participants. Due to missing data from several individuals (e.g., the eye-tracker captured zero fixations) only participants with more than 70% of non-zero data from the eye-tracking phase

$\boldsymbol{\theta}$	Adults	Infants	p
α	0.22	0.11	10^{-2}
β	3.72	3.41	10^{-1}
ϕ	3.37	1.94	10^{-11}

Table 5.1: Parameter estimates of the TD model with "biased" choice parameter ϕ. Infants appear to have a slower learning rate α than adults in the specific experiment. There was not any significant difference observed between the exploration/exploitation parameter β between the two groups. Finally, adults seem to pay more attention to the actor's actions than the infants, as they have a significantly higher value of the ϕ parameter. The p values were calculated after a t-test for testing the assumption (null hypothesis) that the parameter values for the two groups do not defer. This model performed best according to the Bayesian Model Selection method described in the main text.

were included in this analysis. Our final sample consisted of 37 adults and 22 infants.

The fits of the best model (eq. 5.14, 5.16) are given in table 5.1. As expected, infants appear to be slower learners than adults. This means that they adjust their value functions more slowely than the adults. In terms of exploration/exploitation there was no significant different. From the mean value of the ϕ parameter, we can conclude that adults, compared to infants, are more attentive towards what is happening on the screen, which is also reflected by their faster learning rate.

The type of models used here allow us to question the specific nature of the updating process described above in sections 5.6.3.2, 5.6.3.3; in particular, whether evidence for learning (i.e., updating) during the observation phase reflects extraction of causal relationship between the Effect pair and the effect, as evidenced by participants' ability to reproduce the light effect themselves.

Next, we examined how the model correlates with the behavioral data. We found that there is a significant correlation ($r = 0.33$, $p = 0.05$) between the maximum value of the value function for Action 2 with the conditional probability of performing Action 2 following Action 1 (for the Effect pair). This means that an observer with a higher value for Action 2 was also more likely to produce the entire effect pair during the behavioral phase.

The above statement does not imply that in all cases a person that has high probability of producing the effect pair is aware of the fact that only Action 1 and Action 2 are producing the effect. For example, a participant might produce a longer sequence of actions before producing the Effect pair and thus think that

this particular sequence produces the effect. This was verified by a question regarding participant's explicit knowledge on the combination that switches on the light at the end of the whole experiment. More specifically, each adult participant was asked if they knew how to switch on the light. Most of adults knew how to switch on the light, although, they were not aware of the exact action combination. For example, if the Effect pair consisted of actions bend and open, a participant might answer that the correct combination was push, bend, bend, open. As we mentioned before, the participant did not have explicit instructions to discover the actual sequence that produced the light. This means that they did not explore enough while interacting with the toy-box, rather they preferred, once they found a sequence that worked, to stop engaging with the toy.

5.7 General Discussion

In this study we examined in detail the learning process that takes place during observation of sequential human actions. We used a novel task to test aspects of learning and transfer of knowledge between a phase demonstrating the task and a phase in which a participant was interacting with the task. Although the data were initially collected without a possible computational approach in mind, we successfully applied our modeling techniques to these experimental data sets and extracted some insightful results. We attempted to give computational explanations of patterns observed in the behavior of the participants in light of eye movement patterns observed in the demonstration phase of the task.

As we described in the introduction of this chapter, causal inference is a fundamental component of human reasoning and learning abilities. During development, imitating others' behavior is one important way in which infants can identify important sequential actions that cause desired effects in the world. This reflects a developmental process through which infants can infer structure within their environment.

We implemented a RL framework for modeling which, to the extent of our knowledge, provides a novel approach that differs from that implemented by previous researchers (e.g., Baldwin et al. (2008); Buchsbaum et al. (2015); Meyer et al. (2011, 2010)). First, we examined an online Bayesian learner to test if participants maintain beliefs about actor's action transitions. The simple Bayesian model managed to predict the probability of future eye-fixations on various AOIs.

This, indicated that participants did use statistical inference, according to their beliefs, in order to predict events. The fact that according to our model, only the transitions of the Effect pair and partially of the Non-effect pair were learned, supports our hypothesis on the important role of salient effects or causal effects in learning.

Then, we tested the assumption that the duration of an eye fixation could reveal a participant's preference for a particular AOI. We applied a simple RL approach to the eye-tracking data, interpreting duration of eye fixations as a participant's preferences for particular AOIs, and thus, as values of a value function that was updated by a model-free RL model. The action selection method used this value function in order to predict the most probable eye-fixation location. With such an approach, we managed to build a model that not only predicts the duration of an eye-fixation but also the location of it, in contrast with the Bayesian model which just predicted how probable it was for a participant to look at each one of the toy-box AOIs.

We examined behavioral and eye movement patterns from a different perspective. Assuming that a participant would find it rewarding to predict the actor's next action we proposed that a (internal) reward was received in such cases. Then, we fitted various model-free models to the eye-tracking data. We concluded that the model fits combined with the behavioral patterns (probabilities of turning on the light, section 5.6.5) showed that the model's value function for a particular AOI was correlated with action performance (i.e., how many times a person successfully activated the light) during the toy-box task. In accordance with our main hypothesis, that a salient effect can speed up learning, we found that participants with the maximum value of their value function, for the second action of the effect pair, higher than any other's AOI value function, had higher probability of producing the effect pair and turning on the light than those with lesser values of their value function.

The BAMDP framework provided an integrative theoretical framework for learning via observations. It combines learning in an entirely unknown environment with subsequent planning, by capitalizing on the knowledge acquired during the learning phase. Our aim was to test if prior knowledge about transition probabilities acquired during action observation could be immediately transferred during the action execution phase. However, as discussed the model remains to be tested under more constrained experimental conditions than the ones of our paradigm. A

POMDP framework in which we consider a subject's belief of fixating at a location, similar to the evolving distribution of eye fixations (fig. 5.11), might be more insightful (e.g., Butko and Movellan (2010)). It remains also for future work to evaluate how the information gained by the subject by scanning a particular region could guide learning.

Finally, we concluded that a salient action-effect increased the value of the action preceding the effect. This implies faster learning for the Effect pair. During the phase of real interaction with the toy-box, adults who engaged with the toy-box, without any instruction, successfully activated the light. Infants could not be included into this category as their own action sequences were too sparse to be captured by our models.

Further work could be focused possibly on modifying the experimental procedure by enforcing timing constraints and establishing a goal. This would help participants to focus on turning on the light as many times as possible. Thus the evidence of learning (or not) from the previous phase would be enhanced. Another issue that needs further examination is the learning rate. In our case, we used a common learning rate for all AOIs during the whole experimental phase. It is thus impossible to distinguish inter-subject or group differences regarding the learning rates, which, in the case of developmental research, are particularly interesting as they could reveal the differences between learning rate function among different age-groups. Extensions and further research could include varying step-by-step the learning rate such as in a Kalman filter, which could reveal differences in trial-by-trial learning rates of adults and infants. Another potential direction would be to implement a polynomial or exponential function for the learning rate, and potentially examine the form of this function and the differences among age groups.

5.8 Highlights

In this case study we were interested in developing computational approaches that could explain learning of causal relationships between actions and effects demonstrated in a video. For this, we used behavioral and eye tracking data collected from humans (adults and infants) while engaged in a **two-phase experimental procedure**. In the first phase, the *demonstration phase*, participants watched a video of an actor interacting with a toy-box which was consisted of different buttons and their gaze fixations were recorded. A specific combination of the buttons

(actions) was responsible for a light to turn on (effect). In the second phase, the *interaction phase*, participants were allowed to physically play with the toy-box and attempt to turn on the light.

There were **three computational approaches** followed here to fit the models to the data, and **one theoretical** computational framework was suggested. In the first case we assumed that participants maintained–and updated at each time step– a belief over which action the actor will choose next. The Multinomial-Dirchlet model proposed, a Bayesian model, successfully predicted the participants' gaze location–prediction in the sense of predicting gaze location at time step t based on history up to $t-1$.

In the second case, we assumed that participants learned associations between actions and effect by spending more time fixating at a particular location which was of greater importance for them. That importance for each location was reflected in the value function of each location, which was learned in a Model-free RL way, and was an estimate of the expected duration of an eye fixation for that particular location. The action selection process used assigned a probability distribution over locations which closely matched the empirical probabilities derived from participants' data. The model fitting was implemented by using a maximum likelihood approach. The model revealed the development of strong preferences on choices that led to a rewarding effect.

In the third case we assumed that participants were finding it rewarding if they could predict the actor's next action. According to this scenario we fitted three Model-free RL models: A simple R-W and a TD model with two different action selection processes. Furthermore we attempted to find correlations between the demonstration phase and interaction phase, and explain this computationally. The model parameters were inferred by using maximum a posteriori and from this we extracted the following results:

- Model-free RL methods can:
  - effectively explain the learning mechanism of causal relationships among actions and effect in a stream of sensory information.
  - describe the mechanism with which participants learn from the actor's movements.

- Value functions of the action-effect pair are highly correlated with the likelihood of a participant learning by demonstration and reproducing that effect later on when interacting with the task.
- Infants adjust their predictions much slower than adults.
- Adults show a higher level of attention towards the actor's actions on the screen compared to infants.

Apart from the work that was related to model-fitting we provided a theoretical framework based on Bayes-adaptive planning that can describe how knowledge is transferred from the demonstration phase to a hands-on interaction phase with the task.

Chapter 6

Uncertainty-driven Exploration

ABSTRACT

The dilemma between exploration and exploitation is crucial during action selection. The cost of exploration sometimes might be prohibitive whereas in other scenarios exploration might lead to suboptimal results and interruption of development. In this chapter we investigate the way that adolescents to balance exploration and exploitation during their decision making. We argue that their exploration is triggered by the most uncertain choices presented to them and the more uncertain they are for a specific option the more likely it is to select that option. The mechanism behind this process is mathematically explained by a drift diffusion model which consists of different components, each one accounting for different contributors of the action selection mechanism.

6.1 Background

When observing animals in the wild, they are frequently to be found foraging for food. If successful, they will be rewarded by getting fed, which eventually leads to their sustainability and ultimately survival. Their survival often depends on their foraging strategies. For example, it might be the case that they have to decide between exploiting the resources (reduced at the specific given time) of a well-known area, or exploring a completely new territory. This dilemma, between exploitation and exploration, should be balanced for the survival of the animal. Too much exploitation might lead to the exhaustion of the available area's resources, whereas too much exploration might lead to risks of starvation, or facing other predators, which might lead to the death of the animal.

Similarly, an RL agent, which might represent a simplistic version of an organism (as described in previous chapters) selects an action at every time step t according to a policy, and receives feedback from the environment in the form of reward/punishment. However, to act optimally in an environment the agent has to balance how often it explores (i.e., trying different options) and exploits (i.e., using its knowledge to select an option). The balance between exploration and exploitation is a very critical component of the RL methods. Thus, an agent should attempt to balance the utilization of acquired knowledge with the often risky decision to examine uncertain options.

Evidence from studies suggests that individuals may explore options whose contingencies they are more uncertain of (Dayan and Jyu, 2003). In order to examine the exploration/exploitation trade-off, Moustafa et al. (2008) introduced a dynamic reward-learning task, the clock task, and an associated mathematical model to characterize and predict response times of individuals. The clock task (fig. 6.1), in brief, consists of a clock face and an arm that does a full circle in 5 seconds. Participants need to stop the clock, before the end of its course, in order to gain points that vary according to the latency of their response time (RT). In general participants are faced with choices that lead either to a small immediate reward or that would produce large delayed reward. In other words, the probabilities of reward and the magnitude of reward vary. Therefore, subjects must learn the statistics of reward probability, magnitude, and their integration, as a result of experience across multiple trials within a given context, and adjust their response time (RT) accordingly.

Initially, the clock task was used by Moustafa et al. (2008) to test whether striatal dopamine (DA) increases enhance "Go-learning"[1] to pursue actions with rewarding outcomes, and DA decreases enhance "NoGo-learning" to avoid non-rewarding actions. This hypothesis was tested with Parkinson disease patients. The main idea was that accumulated positive reward prediction errors drive basal ganglia-dependent Go learning to speed up responses, whereas negative prediction errors have the opposite effect.

In their analysis, a neural network model of the basal ganglia (Frank and Claus, 2006) was used, which simulates the experiment. Specifically, it simulates systems-level interactive neural dynamics among corticostriatal circuits and their roles in

[1] A very good explanation of the basal ganglia pathways and Go and NoGo learning can be found in Frank (2007).

action selection and reinforcement leaning. The model also accounts for various effects of dopaminergic manipulation on action selection and reinforcement learning. In their simulations, they used parameters already estimated from previous experiments. Thus their results can be considered as predictions rather than model fits to new data.

It was found that although participants were not optimal, they nevertheless learned to adapt RTs in the direction expected. Their tendency to adapt response times to maximize expected reward was found to be dependent on dopaminergic medication status. While they were off dopaminergic medication, patients showed slower responses to avoid early low expected values, but were less able to speed up when their early responses were rewarded. The opposite was observed when patients were on medication. Their speeded response was better whereas their response slowing became worse. The same patterns were also observed by their model of basal ganglia. They concluded that, according to their experimental and computational data, the striatal dopamine effects on decision making and probabilistic selection paradigms tap into common mechanisms.

Frank et al. (2009) used the same task and mathematical model to study the neurogenetic contribution to the exploration/exploitation trade-off. Specifically, they showed that genes controlling striatal dopamine function (DARP-32 and DRD2) are associated with exploitative learning to gradually adjust participant's response times as a function of positive and negative decision outcomes. On the other hand, a gene responsible for mainly controlling prefrontal dopamine function (COMT) was found to be associated with exploratory decision making, in which decisions are made in proportion to relative uncertainty about whether other alternatives might yield outcomes that are better than a given status quo.

In a similar study, Cavanagh et al. (2011) used EEG data to investigate the interaction of middle and lateral frontal areas during top-down strategic control involved in exploratory choices. It was found that theta-band activities reflect prefrontal-directed strategic control during exploratory choices. In particular, midfrontal and right lateral/frontopolar areas seemed to "track" the degree of relative uncertainty of the chosen option (as compared to unchosen option) up to 500ms prior to response commission. The model parameters disambiguate the likely roles of the EEG patterns observed, especially EEG in the theta band range, for which there was a correlation with various features of reinforcement learning such as unexpectedness updating and decision uncertainty.

Badre et al. (2012) used the clock task and the same modeling approach in a fMRI study. Results indicated that rostrolateral prefrontal cortex tracks trial-by-trial changes in relative uncertainty (e.g., uncertainty about which response, fast or slow, yields a positive reward prediction error), and this pattern distinguished individuals who relied on this uncertainty for their exploratory decisions versus those who did not.

The above observations have also been verified in the developmental literature. For example, Schulz and Bonawitz (2007) suggested that preschooler's exploratory play was sensitive to stimulus features such as novelty and perceptual salience. Bonawitz et al. (2012) reported evidence in support of the claim that children's learning is conservative and flexible, and that they integrate evidence, prior beliefs and causal hypotheses into their exploration. Thus, children seemed more likely to explore when they observed evidence that conflicted with their prior beliefs than when they were presented with belief-consistent evidence.

The adolescent/pre-adolescent period is considered as a time when there is an increase in novelty-seeking and exploration behavior. Our concern in this chapter is to investigate if children also balance exploration and exploitation. In other words, do children select their actions based on outcome expectations or do they prefer to explore new alternatives that might result in more rewarding contingencies? To examine this, we use the analysis of Badre et al. (2012) and the experimental paradigm of Moustafa et al. (2008), in which subjects stop a rotating clock in order to win points, to examine if the behavior of pre-adolescent children is also consistent with the uncertainty-driven exploration strategy reported by Badre et al. (2012). Specifically, we use data collected from 45 (10-year-old) pre-adolescents tested under the same conditions as in the original study. All experimental procedures were carried out by Ezgi Kayhan at the Donders Institute of Radboud University in Nijmegen (Netherlands).

6.2 Task - Experimental Procedure

We used a task that has previously been used to investigate trial-specific reinforcement learning and exploration in genetic, patient, and pharmacological studies (Badre et al., 2012; Cavanagh et al., 2011; Frank et al., 2009; Moustafa et al., 2008). Participants were presented with a clock face (fig. 6.1) whose arm made a full turn in 5 seconds per trial. They were informed of the amount of time the arm

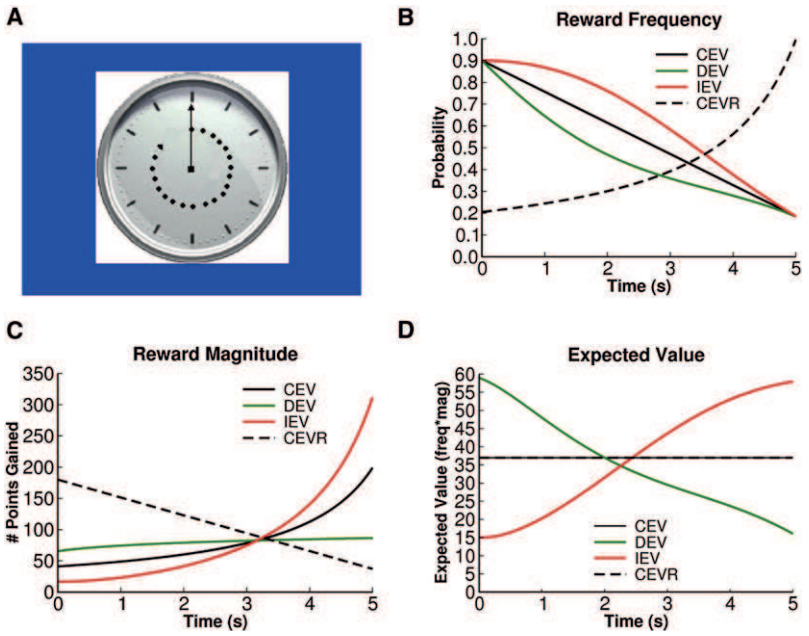


Figure 6.1: The Clock task and the reward function conditions. **A.** The clock face task in which participants have to stop its arm within 5s, else the trial is ended and considered unsuccessful. **B.** The probability of reward as a function of response time in seconds, for all four conditions. **C.** The magnitude of reward across time for the four conditions. **D.** The expected value of reward (reward magnitude $\times$ probability of reward) as a function of response time. Both CEV and CEVR conditions have constant expected value but in the CEVR the reward magnitude decreases over time whereas the reward frequency increases. (Figure adapted from Badre et al. (2012))

needed to make one full circle, and were asked to press the 'spacebar' button before the arm finished its turn in order to win points. If they did not respond within the 5 seconds, they did not win any points.

The duration of the experiment was fixed, no matter how fast the participant responded. The points earned in each trial were determined by when the participant responded and the current condition of the clock. There were in total four conditions, comprising 50 trials each, in which the probabilities and magnitudes of rewards varied as a function of time elapsed on the clock until the participant responded. Before each new condition, participants were informed that a new clock would appear and, in general, they were encouraged to try different response times in order to learn how to gain the most points.

The four conditions were defined according to the expected value of the reward, $EV = \mathbb{E}[r] = \sum p_r \cdot r$, which is equal to the probability of the reward occurred multiplied by its magnitude. The three main conditions, dependent on the EV, were: **i)** CEV (constant EV), **(ii)** IEV (increasing EV) and **(iii)** DEV (decreasing EV). Within each trial in the three conditions, the number of points (reward magnitude) was increased, whereas the probability of receiving the reward was decreased over time. In the fourth condition, **(iv)** CEVR = CEV Reverse, the expected value was constant but the reward magnitude decreased and reward probability increased as time elapsed. This condition was included for multiple reasons.

First, as both CEV and CEVR conditions have equal expected values across time, any difference in subjects' response time (RT) might reflect potential bias to learn more about the reward magnitude than the reward probability or vice versa. For example, waiting longer in CEVR than in CEV might indicate risk aversion because the participant values more frequent rewards than higher magnitude rewards sparsely received. Another reason for introducing the CEVR condition is the possibility of disentangling whether trial-by-trial RT adjustment effects reflect a tendency to change RTs in the same direction after gains, or whether RTs might change in the opposite direction.

The order of conditions (CEV, DEV, IEV, CEVR) was counterbalanced across participants. After every 50 trials a small rest break was given. Although participants were instructed to win the most points, at the beginning of each condition, they were not aware of the different conditions. However, the color of the clock face was changed depending on the current condition. To avoid memorization of associations between clock face color and timing-reward, a small amount of random

uniform noise was added to the reward magnitudes at each trial.

6.3 Computational Model

To model the trial-by-trial response times, Frank et al. (2009) and Badre et al. (2012) used a RL-based model with several of different components. In this section we describe step-by-step the assumptions of the main equation, with which the RTs for each participant for each trial was estimated, was based:

$$\hat{RT}_t = K + \lambda RT_{t-1} - V_t^{Go} + V_t^{NoGo} + \rho(\mu_t^{slow} - \mu_t^{fast}) + \nu(RT_{best} - RT_{avg}) + Explore_t \tag{6.1}$$

where K is a baseline response speed, λ autocorrelation with the last's trial RT. This is the equation that was used to estimate the response time of a participant. It is composed by several quantities that we will analyze in detail in the following paragraphs.

As we mentioned, the core of the eq. 6.1 is a simple RL model. Therefore, the central assumption was that participants maintain and update in every trial an expected value for the reward that they expect to gain in trial t:

$$V_{t+1} = V_t + \alpha \cdot \delta_t \tag{6.2}$$

where α denotes the learning rate, or in other words, the amount by which the difference $\delta_t = r_t - V_t$ (reward prediction error or RPE) between actual r_t and expected reward V_t affects the prediction of the value in the next trial $t+1$. This is the standard R-W rule described in previous chapters. According to this, the expected reward is updated at every trial according to sampling experience from each experimental trial. The expected value in eq. 6.2 encodes two separate mechanisms for approach-related speedy responses ("Go-learning"), presumably caused by accumulated positive prediction errors, and slowed responses ("NoGo-learning"), presumably caused by negative prediction errors, according to eq. 6.2:

$$V_{t+1}^{Go}(s,a) = V_t^{Go}(s,a) + \alpha_G \cdot \delta_t^+$$
$$V_{t+1}^{NoGo}(s,a) = V_t^{NoGo}(s,a) + \alpha_N \cdot \delta_t^-$$

where a denotes the action taken (slow or fast response in comparison with an average response) and s is the state of the clock face. In accordance with Moustafa

et al. (2008), where they used a Neural Network model representing the function of the Basal Ganglia, there is an explicit separation between Go Learning (learning to reproduce behaviors that yield positive outcomes) and NoGo Learning (learning to reproduce behaviors that yield positive outcomes). In particular, Go learning is facilitated by the action of (excitatory) D1 receptors in the striatonigral pathway whereas the NoGo learning is facilitated by the action of (inhibitory) D2 receptors in the striatopallidal pathaway. Thus, the agent might learn different information from positive prediction errors (PPEs) than negative prediction errors (NPEs) (Frank et al., 2009).

The first assumption is that subjects maintain a belief over how fast or slow they should respond on the next trial in order to have a positive prediction error (obtaining a better than average outcome). Thus, participants simply adjust their RTs in proportion to the difference between their expected and achieved reward values. The beliefs over fast/slow responses are updated online according to Bayes rule:

$$P(\theta|\delta_1,\ldots,\delta_T) \propto P(\delta_1,\ldots,\delta_T|\theta)P(\theta) \tag{6.3}$$

where θ is the parameter of the belief distribution about the reward prediction errors $\delta_{1:T}$ observed from trial 1 till trial T. In our case, considering that participants track positive RPEs, we can assume a Binomial likelihood, $\text{Bin}(n|N,\theta)$, representing the distribution over how many times n a positive RPE was encountered in N trials. As it is common in Bayesian approaches, we assign a $Beta(\theta|\eta,\beta)$ distribution as prior over the Binomial's distribution parameter θ. Eq. 6.3 describes how the posterior distribution over θ (which because of the conjugacy between likelihood and prior will be also a Beta distribution) updates over time, according to the encountered RPEs. It is important to clarify that this framework is an assumption on how a participant might change his or her beliefs about obtaining a better than average outcome and it is illustrated in fig. 6.2.

The learned expected values of fast/slow responses (as the means of two Beta distributions modeling positive RPEs for fast/slow responses) that contribute to the estimation of the RT is the following term:

$$\rho(\mu_t^{slow} - \mu_t^{fast}) \tag{6.4}$$

where ρ is a scaling factor of the difference of the reward statistics. Similarly,

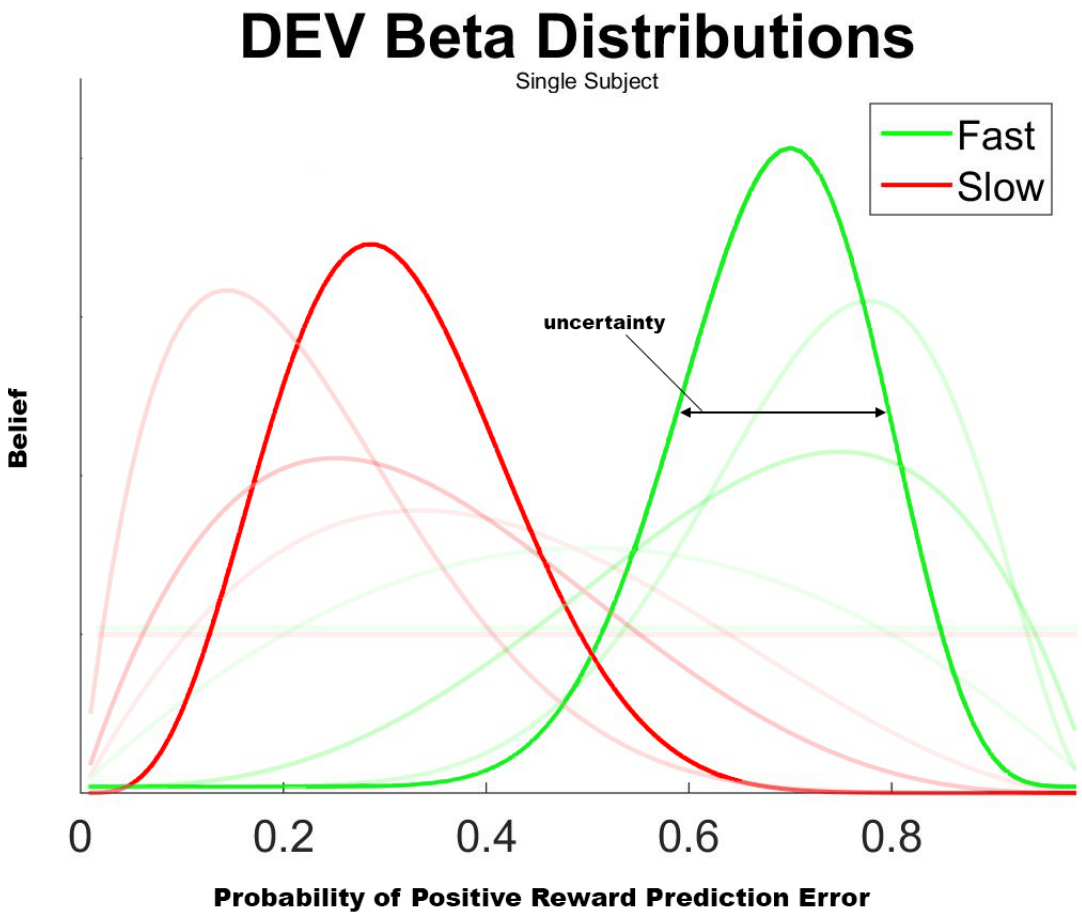


Figure 6.2: Evolution of a participant's belief updates over fast or slow responses, during the course of the experiment, under DEV condition. The x-axis represents the probability that a particular action, slower or faster response, will result in a positive prediction error. The y-axis represents the level of belief that a participant has about each probability for responding faster or slower (compared to an average response). Exploitative responses will move to the direction of the highest perceived value of a particular option. In the DEV case, the expected value is decreasing therefore the subject's beliefs are evolving favoring the faster responses, starting with equal beliefs, as these responses are more likely to yield a positive reward prediction error. The standard deviation of each distribution represents the participant's level of uncertainty, regarding the value of the corresponding option. Thus, early in learning the uncertainty is larger and later smaller. The difference between the fast and slow standard deviations, at any given trial, reflects relative uncertainty.

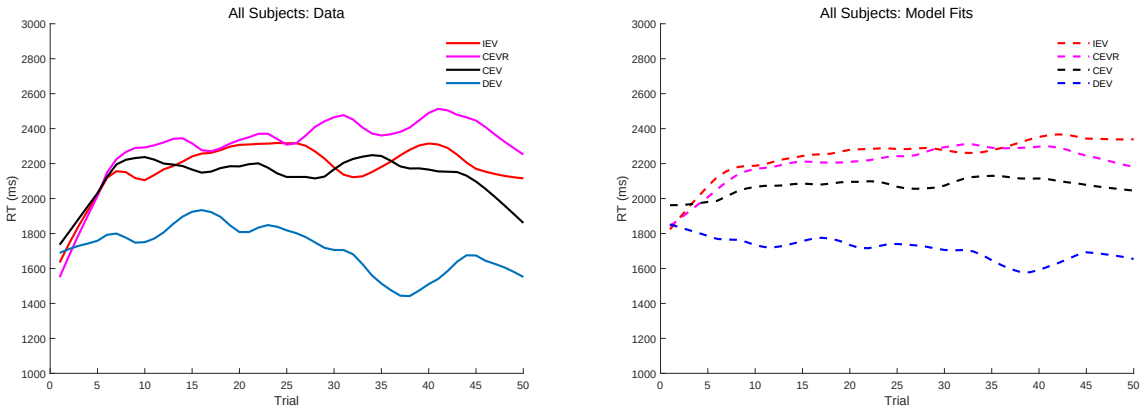


Figure 6.3: Subjects data patterns and model generated data.

another term that was presumed important is:

$$Explore_t = \epsilon(\sigma_t^{slow} - \sigma_t^{fast}) \tag{6.5}$$

where σ^{slow} and σ^{fast} are the uncertainties of the Beta distributions. The scaling factor ϵ was constrained from below by 0. This component was named "Explore" as it drives exploration towards responses for which reward statistics are most uncertain.

To summarize, model fits provide subject-specific, trial-by-trial estimates of reward prediction error (δ+, δ-), the means of the Beta distributions (μ^{slow}, μ^{fast}) and their corresponding standard deviations (σ^{slow}, σ^{fast}), which represent the uncertainty over positive RPEs. Another estimate provided by the model is the participant's reliance on relative uncertainty to explore ϵ.

6.4 Results

We tested 44 10 year old children. In fig. 6.3 we present the average response time over all participants per trial per condition. Participants seem to adjust their incremental RTs in a way that is consistent with learning. For example, a subject being tested under the IEV condition (increasing expected value of reward) responds on average more slowly whereas under the DEV condition tends to respond faster.

In the constant EV conditions, the 10-year-olds seem to weigh the frequency of the reward more than its amount. This can been seen from their response times, as they respond faster in the CEV case rather than the CEVR case. It might be

the case that they prefer frequent reward as it is distributed across trials, under CEV and CEVR conditions.

In general, we can observe that participants learn to adjust their response times according to the amount of reward they receive (e.g., in the DEV condition RTs are faster because the amount of reward received decreases with respect to time). The fitted model parameters can be used to simulate the response time of an artificial agent performing in the clock task. From each condition we can plot the RT adjustments as in fig. 6.3. The model manages to capture the average patterns of the data but fails to capture the RT adjustments in IEV and CEVR case especially at the last trials.

We also present data from an individual in fig. 6.5 across 50 trials for each condition. Again, it can be seen that the subject adjusts his or her RTs, indicating learning. For example, for the Subject no. 3, he or she begins with fluctuations during an exploration phase and then adjusts his/her reaction time faster as the EV decreases over time.

It is important to note that under all conditions, there is an exploratory phase (usually some initial trials) where the subject tries different options. In the CEV condition for example, Subject no. 3 attempts slow responses at the beginning, which in this specific condition are not rewarding, then he or she tries faster responses and finally adjusts to slower responses (fig. 6.4). However, in both CEV/R conditions, there are lots of fluctuations as the EV is constant per trial, which is hard for the participants to distinguish.

In fig. 6.5, we present the exploration parameter ϵ (restricted in this study to take only positive values) and how it behaves compared to the RT swings ($RT_t - RT_{t-1}$), which indicates how a participant might adjust his or her RT according to the amount of exploration he or she uses. Usually, the larger swings are observed when the exploration parameter has a high value.

We found that the correlation between RT swings and relative uncertainty (difference in deviations of fast/slow responses) of explorers was significantly different than zero (mean r=0.28 $p \ll 0.001$, t(19)=14). This is similar to Badre et al. (2012) results (r=0.36, $p < 0.0001$) for adults. This leads to our conclusion that 10-year-olds decision making patterns are consistent with the hypothesis of uncertainty-driven exploration. From the scatter plot fig. 6.6, it can be seen that the more uncertain about a choice, if it yields a positive RPE, a subject is, the higher the adjustment of the response will be.

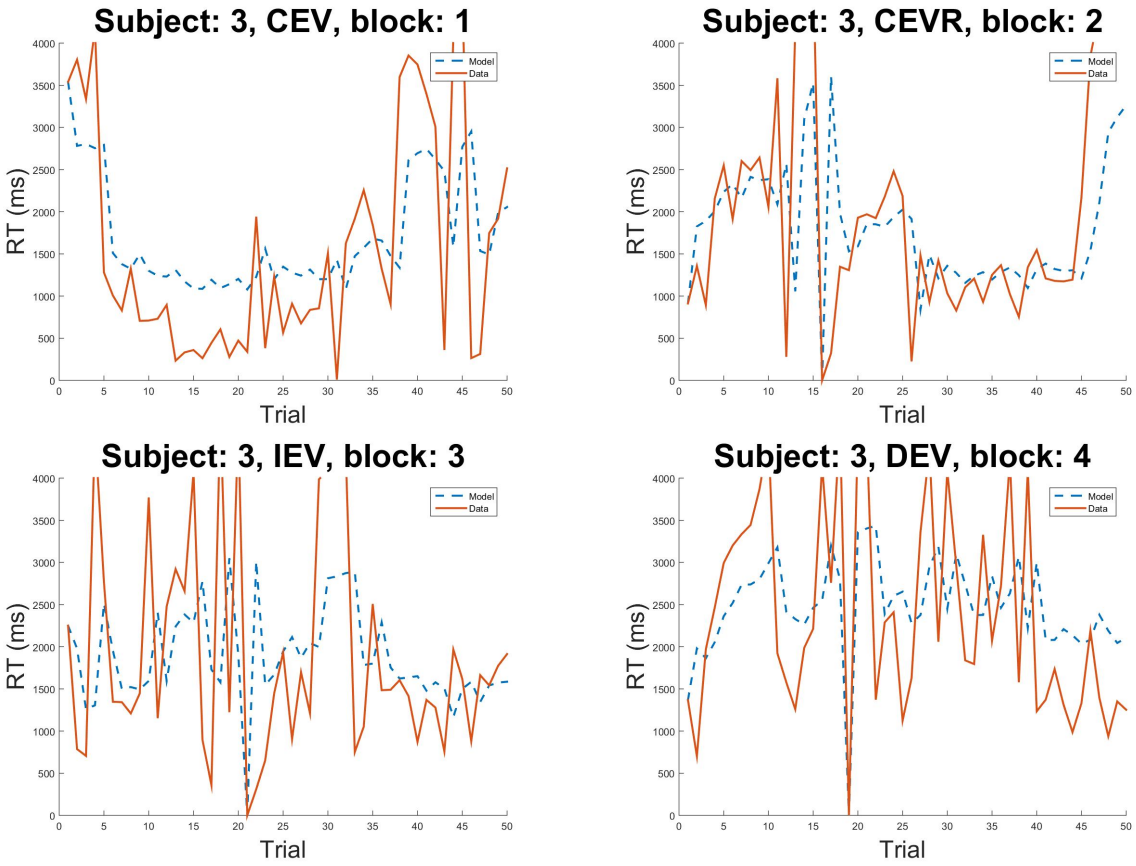


Figure 6.4: Single subject's response times in four conditions. Each figure presents a subject's time responses along with the model's predictions for each one of the four conditions (CEV/R, IEV, DEV).

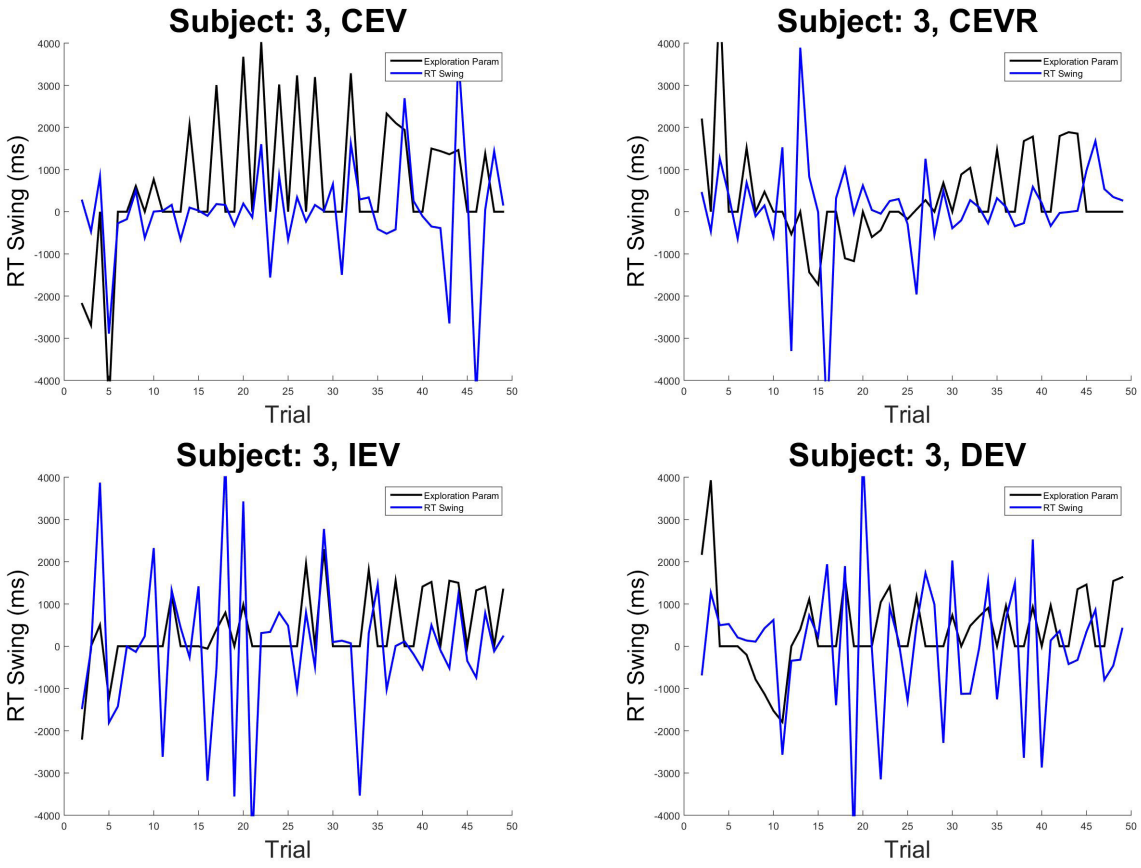


Figure 6.5: A single Subject's RT change and estimated Exploration parameter ϵ in the four conditions. The RT swings denote the difference between current RT with the previous trial's RT. Exploration parameter seems to capture paritally the trial-by-trial RT differences.

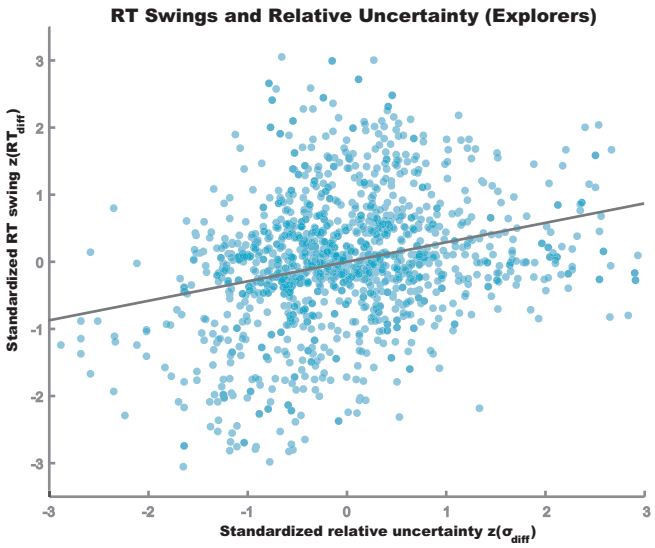


Figure 6.6: Correlation between RT swings and relative uncertainty among explorers. Data points denote a single trial from a participant and th e plot includes all data from all trials and participants.

6.5 General Discussion

We adopted the modeling and experimental methods of Badre et al. (2012) to replicate their analysis with data from 10 year old children. Our main goal was to test whether their hypothesis (exploration is driven by choices that participants are most uncertain of) applies also to different age groups. We found out that indeed there is a correlation between children's response times and the relative uncertainty. This is in accordance with the main hypothesis. Badre et al. (2012) reports that, in their experiment, non-explorers had a correlation between the relative uncertainty and their response times that was not significantly different from zero. It is left for future work, to investigate the behavior of the non-explorers. In addition, a comparison of the amount of exploration used by children and adults might provide useful insights on how uncertainty might affect exploration.

The task seems also appropriate for younger children as there are no planning mechanisms involved. This is because the whole learning process is an online process which is updated by sampling experience (i.e., trying different timings for stopping the clock arm). In addition, there is no need for any training sessions prior to the experimental phase as is necessary in planning tasks (i.e., learning transition probabilities and reward probabilities). We believe that such a task could therefore be useful for examining model-free RL approaches in young children

without the involvement of planning procedures. Furthermore, the digital nature of the task helps the exact replication of the experimental procedure, which helps the examination of different age groups for developmental studies.

In terms of improving the model, after personal communication with the second author of Badre et al. (2012), (Michael J. Frank), there are not many things to be done. Eq. 6.1 was built gradually, adding one component at a time, and every time the performance of the model was evaluated. The model was also compared with a Kalman filter. In addition, the model has been tested also with Gaussian beliefs instead of the Beta beliefs we used here. In all cases the current model outperformed all other versions of it. However, as we mentioned the task and model provide a good framework for testing different age groups and the analysis can also integrate additional data from EEG or fMRI. We note that Michael J. Frank reported that his team was working on that direction.

6.6 Highlights

In this case study we were interested in examining the hypothesis that humans explore choices that are more uncertain of, in 10-year old population. The whole process follows Badre et al. (2012) with similar results verifying our main hypothesis. The model used to estimate the response time was a mathematical model that consisted of many different components. These components were updated at every time step. The model fitting was implemented with a built-in function of MATLAB for unconstrained optimization.

The main contribution of this chapter could be summarized below:

- Using a compositional mathematical model we verified that adolescents choose the actions that are less certain for the outcome while exploring different options in a given task.

Chapter 7

General Conclusions

ABSTRACT

In this chapter we conclude the whole thesis. We refer to the main contributions and review the effectiveness of the general computational approach used throughout each project. Furthermore, we stress the limitations of the methods used, possible solutions, and discuss future research directions.

7.1 Contributions

The work presented here was carried out with the aim of demonstrating the advantages of Machine Learning approaches, specifically Reinforcement Learning methods, for explaining behavioral patterns. The aim of the thesis was to illustrate that RL methods can serve as a theoretical and computational framework for various aspects of decision making in the context of human action selection, and specifically with respect to data from cognitive development. It is hoped that these methods could be useful for understanding existing experimental data and also analyzing future experiments.

The core of this thesis lies in the fact that many decision making problems can be formulated into MDPs which can be solved by the algorithms described in Chapter 2. The solutions are categorized as model-based or model-free, depending on the availability of a model of the environment, which consists of a reward distribution and a distribution over the transitional dynamics. This categorization has behavioral realizations, and provides a good theoretical framework for goal-directed and habitual types of behavior.

It is well known (Daw, 2012; Daw et al., 2011) that animals and humans em-

ploy heuristics to provide a cheap approximation to the amount of exploration-exploitation they need, in order to make a decision. This mechanism is related to mental planning or cognitive search. Pfeiffer and Foster (2013) showed that before goal-directed navigation in an open arena, spatial representations of neuronal firing patterns, in the area of hippocampus encode future spatial trajectories strongly biased to progress from the subject's current location to a known goal location.

Whereas goal-directed behavior is regulated by the action's outcome, it is also common for human or animal responses to turn into habits (not sensitive to outcome devaluation), after long exposure to the same stimulus. This kind of behavior is considered reflexive and elicited by stimuli rather than their consequences. An action is selected mostly based on its reward history rather than its consequences. We showed how each of these behaviors can be explained by the RL framework, which variations of RL algorithms are appropriate, and why.

In Chapters 2 and 3 we illustrated how behavior and learning could be formulated as an MDP and solved by RL algorithms. We demonstrated how RL models can fit behavioral data from a simple decision making experiment. Finally, we presented a Bayesian scheme for model parameter inference and we applied the above methods in a simple spatial-navigation domain.

In Chapter 4, our concern was whether intrinsic motivation might play a role in the cognitive processes underlying planning. Intrinsic motivation reflects the drive of cognitive search from the initial state to the goal state, in cases where there is reward only at the goal state. We assumed that this motivation emerged in the form of an additional *reward shaping function* (Ng et al., 1999), and behaviorally as a perceptual strategy, which guides the subject towards moves that bring the current configuration of the task "closer" to the goal configuration, no matter if these moves lead to a non-optimal longer solution path. For example, in the ToL task configured as in fig. 4.4, a seemingly "natural" move (moving the red ball to the shortest peg) might not be optimal for the overall goal of the task. Comparing the goal state with the start state, moving the red ball to the shortest peg, brings the start state to a state more similar to the goal state (red ball will be at the same position as in the goal state, and green and blue balls will be at the same peg as in the goal state, but with reverse order). This observation indicates that there might be a separate component that affects the mental planning mechanism.

We fitted three modified model-based RL models to the data collected from children (3-to-4-year-olds and 5-to-6-year-olds) playing the ToL task. The modi-

fied reward function enabled the models to capture how much children are affected by the state similarity towards their attempt to solve the tasks. We showed the developmental role of the reward weighting parameter and how this changes across age, revealing that younger children had a tendency to use the perceptual strategy more often. Finally, we concluded that the planning process is affected by a perceptual system that biases the choices of a person to particular branches of the decision tree and that this was observed mostly in young children.

This perceptual strategy might emerge because of the particular features of the task. There are many examples in board games that demand great use of the mental planning system, like chess. In such games it is often observed that player A attempts to force the other player B into particular moves that seem initially beneficial for player B, but ultimately benefit player A. These types of strategies attempt to bias the opponent's decision.

In the computerized version of ToL we used examples of such stimuli, the goal state, that seemingly provoke the player to make the 'wrong' moves. Because of the nature of the task, in which a player usually chooses sequential subgoals (i.e., to put a ball exactly at the position of the goal state), particular combinations of start and goal states lead the participant to choose the non-optimal subgoal first. Surprisingly, in the computerized version of ToL, tested with adults, the pruning models with our reward modification accounting for the perceptual strategy, did not perform better than simple pruning models. However, as discussed, participants were explicitly asked to plan before acting, a process that probably inhibits the online perceptual strategy. Perhaps, allowing the participants to attempt to solve the task freely might result in the similarity strategy being engaged more.

Finally, we introduced the Planet Task which would enable us to examine in detail how the pruning process might work. The task was designed such as to provide a reward for every single action which a participant might choose. Initially, every participant was trained to learn the state transition map of the task and the reward distribution. Unfortunately, the experiment was not successful for reasons that have to do with the suitability of the task for the specific age group that was tested, along with the experimental phase (i.e., the child did not want to continue playing, thus he or she did not manage to complete enough trials to fit the model to the data). However, we extracted useful insights about the experimental design especially for developmental research that involves young children and RL modeling methods.

In Chapter 5, we used a novel task to investigate how humans might learn from an initial video demonstration of the task, and if there was any transfer of knowledge to a subsequent actual interaction task. The task consists of a toy-box with different types of buttons, where only a specific pair of buttons, pushed in the right sequence, turned on a light that lay in the middle of the box. When participants were presented with the video (observation phase), they observed an actor carrying out various actions (including the ones that turned on the light). Afterwards, they had the chance to interact with the toy-box and attempt to switch on the light by themselves (interaction phase).

First, we modeled the eye tracking data from humans during learning from the video phase with model-free approaches. Specifically, we used a model-free RL algorithm to model the learning of the actor's movement transition probabilities that takes place during the observation phase. After presenting the participant's transition probabilities matrices in comparison with the actor's matrix, we discussed theoretical/algorithmic approaches to how these learned probabilities can be used as prior knowledge in the interaction phase. We introduced the BAMDP framework and analyzed in detail possible solutions in order to integrate the planning and learning processes in the observation and interaction phase. We argued that this framework is suitable for the particular task as it describes planning in an environment in which there is uncertainty about the state transition probabilities.

Then, we proposed a model-free RL model to track duration of participant's eye-fixations. That model uses a simple Q-learning algorithm to estimate expected duration of eye-fixation for a particular area of interest (AOI), and used this estimation in order to calculate the probability of fixating on an AOI. The whole process emulates the function of an eye-gaze controller. Our main assumption was that the amount of time spent fixating on a specific area, along with the selection of the area, might indicate a decision making process. Our model predicted probabilities of locations of eye-fixations that are consistent with the real patterns observed, supporting the hypothesis that infants can learn from a stream of actions presented to them.

We further, our investigation by formulating the reinforcement learning problem differently: we assumed that every time a participant predicts the actor's next action, he or she receives a reward. In this context, we fitted different TD models, with parameters that can give us insights into the subject's behavior. First, we found that participants learn the transitions that preceded the effect (i.e. switching

on the light) better than other transitions. Second, we found that infants had a slower learning rate than adults in the observation phase. This indicates that infants were adjusting their predictions on actor's movements more slowly than adults. Furthermore, we reported that adults seem to pay more attention to what the actor is doing than infants, according to the values of the corresponding model parameter.

Finally, we linked the eye-patterns with the behavioral patterns, extracted from the data from the observation and interaction phases of the task respectively, by examining the correlation of the value function of the TD model (that models the eye-fixations) with the conditional probability of producing the effect in the interaction phase. It was revealed that with the RL framework we could predict if a participant could switch on the light according to his or her eye-patterns.

In Chapter 6 we examined if 10-year-olds exploration strategies were driven by uncertainty about their choices. We used a task in which a participant had to stop the arm of a clock in order to win points. The participant's response time defined the reward. This enables one to capture characteristics of an online learning process. Using an RL based mathematical model we found significant correlation between the response times and the relative uncertainty they had on how fast or slow they should act.

In general, attempting to model functions which correspond to cognitive mechanisms with computational methods, eventually might lead to two options: models for which the interest focuses on simulating or reproducing observed behavior – in other words, attempting to find the best approximation of the function being modeled – and models which, although they attempt to simulate the observed behavior, the focus is on the cognitive interpretations of their parameters (Luce, 1995).

The modeling framework adopted and used throughout the whole thesis belongs to the second category. The particular parametrization we utilized is inspired by cognitive neuroscience and knowledge from psychological behavioral patterns. Such a parametrization is not only plausible but defines explicit structural constraints on the mechanisms underlying decision making, thereby providing a quantitative explanation of them. Eventually, the greatest contribution of a computational theory is the ability to quantitatively predict outcomes, which can aid to mechanistically explain and predict behavior.

7.2 Limitations

There are important aspects of behavior, such as hierarchical structure, which we have so far neglected. In the ToL task, a specific problem can also be solved by decomposing it into smaller problems, and then solving each one separately. Under that approach, the subject initially would select a subgoal, then would attempt to bring the ToL into that subgoal state, assign then a new subgoal and repeat. For example, in the problem in fig. 4.4, the subject is faced with two subgoals: i) place the red ball at the shortest peg or ii) place the green ball at the goal state's location. These kind of solutions involve planning over sequences of actions rather than individual actions. In our methods we did not use any hierarchical approaches. However, the amount of hierarchical structure (if any) that young children might use is a question for future research.

A limitation of the child ToL data is that many children broke the task rules by using both hands or placing balls on the table. The computerized version of the experiment provides additional constraints on the participant's choices, which creates better conditions for modeling. Furthermore, the data collected are in digital format which further helps the processing and analysis of them and sharing them between scientific communities. However, phenomena observed in real situations (such as the tendency of picking up the balls in ToL) cannot be observed in such a constrained digital environment. This is simply because the participants do not face exactly the same experimental conditions in the digital version of an experiment. In the non-digital version of the ToL, there are many questions that need to be answered and cannot be addressed in the physical version of it.

In the digital version, young participants are 'forced' to plan whereas in the digital version they use different strategies, attempt to break the rules, etc. For example, the rule violation might indicate poor planning but also a tendency to solve the problem backwards (from goal state to start state). In general, if we want to examine the mechanisms that are involved or affect the planning process, we should also relax relevant constraints. In our opinion, apart from carefully selecting a task to capture a particular behavior, all aspects of the natural behavior a subject might have, in the physical version of a task, should be considered and taken into account in the digital adaptation of the experiment.

Furthermore, instructions for each experiment are equally important as the experimental design, and affect the behavior that a participant might adopt. In-

structing the participants in the ToL task to plan their moves before acting might led them to force a planning strategy, even when this was not their default strategy. This brings difficulties when the purpose of the experiment is to capture the general strategy that each participant employs (and not the planning mechanism per se), such as a mixture of planning and model-free approaches.

Another important issue that rises in the experimental design is the suitability of a task for a particular age. When we use adult subjects for an experiment, the range of tasks, and the difficulty of each one, can vary a lot. However, when dealing with children, we need to pay attention to the suitability of the task for the particular age of the children. Even when a task and the subsequent collection of data in digital data form that favors the modeling procedures seems appealing, it is very hard sometimes to predict if the child can carry out all block of trials. Thus, the the Planet Task ultimately proved to be unsuitable for young children, although it is a very useful task to test the pruning process in adults.

Although the model-free algorithm for estimating expected duration of eye-fixations on particular AOI performed well compared to the behavioral patterns, according to our intuition, we believe that state space models (e.g., Kimura et al. 2010) would perform better (e.g., a Kalman filter or a particle filter algorithm). The reason for this is that RL models are not usually used for tracking. It is more common to use a reward of 1 or 0 (or a punishment of -1) if an agent completes successfully a task or not. In our case, we used as reward the duration of an eye-fixation. Actually, we oversimplified the problem as we did not consider any kind of elements that affect the process of tracking of each eye-fixation's duration and the uncertainty of the estimation. A dynamical model, such as the one used in Chapter 5, modified appropriately, might outperform ours and also give more insights into the control of eye-movements. Furthermore, instead of using discrete locations (AOIs) we could use eye-fixation coordinates, along with state space or dynamical models which estimate continuous quantities. Other approaches that have been successfully applied into the field of control of eye-movements, and that we neglected, are the approaches of Butko and Movellan (2010) and Najemnik and Geisler (2005) (POMDP model and Bayesian model respectively).

The toy-box task is not suitable for testing algorithms such as the BAMCP and MCTS as it is very difficult to extract features from the task. In addition, the parameters of such models do not provide any cognitive links. Furthermore, as we discussed, the experimental procedure could be designed with better constraints

to create better conditions for modeling. For example, at the interaction phase, participants did not have a clear instruction about what to do with the toy-box, nor did they have a time limit. This massively affects the quality of the behavioral results. As we mentioned before, Monroy et. al (2015a; 2015b) did not have any intentions concerning modeling at the time that the toy-box was created and tested.

Despite the model's limitations and sometimes the problematic experimental design, our main goal was to attempt to approximate different types of decision making processes in human brain. The oversimplifications made in many circumstances help to create the necessary conditions in order to describe a problem using mathematical formulas. This enables us to run our computational experiments and test our models and hypotheses. Furthermore, it aids replication of the experiment by the scientific community. Finally, it will be very important in the future to attempt to design experiments so that computational modeling is feasible, and also to attempt to store the data collected in digital format.

7.3 Questions for Future Research

We discussed in detail model-based methods to model human behavior in a planning task. As discussed, there is evidence from behavioral data of various experiments (e.g. Goel and Grafman (1995); Goel et al. (2001); Newman et al. (2003); Simon (1975)) that apart from a planning process, decision making is affected by a perceptual strategy that drives moves. It might be in the form of a complete separate computational component with its own mechanisms. It might be the case of a perceptual bias affecting the decision tree. Either way, this element is formed because of the current environmental state. This type of a system has many implications for the decision making system. Thus, our treatment of this was novel. Eventually, this might lead to a perceptual model embedded in the decision making mechanism. Further questions concern how these systems interact, which features are critical, etc.

If we think of the problem in the ToL task, configured as in fig. 4.4, in terms of subgoals, the options that a participant faces are to either make the appropriate moves to place the green ball at the same location as in the goal state or to place the red ball at the shortest peg as in the goal state. However, young children are more prone to choose the second subgoal rather than the first. This, apparently, indicates that the mental planning process, if represented as a decision tree, is

affected by the characteristics of the task in the form of a strong bias towards specific branches of the tree.

This kind of bias might originate from a different system. It seems also that it is affected mainly by some perceptual characteristics of the task. However, these are questions for future work and investigation. Gershman et al. (2010) worked in that direction, by using a model-free RL model that takes account of the features of the task (such as colors and shapes presented as stimuli). However, that work used model-free RL. Guez et al. (2014b) used task features integrated in the BAMCP algorithm that uses planning. To our knowledge, there are no studies linked to Cognitive Neuroscience that incorporate task features into the planning process using model-based RL methods.

As we discussed in Chapter 4, intrinsic motivation was described by the modified reward function, which incorporates task features. However, there is another framework describing intrinsic motivation and linked to RL methods, specifically, the framework of Chentanez et al. (2005), who developed a different approach of intrinsic motivation using Hierarchical RL. Recently, Kulkarni et al. (2016) used the same concept in Hierarchical Deep RL, and Bellemare et al. (2016) (Google Deepmind) managed to link intrinsic motivation with automatic subgoal discovery. They reported enhanced performance of their agents in the Montezuma Atari game, a game which the previous simple Deep Q-learning agent could not solve. In this game, the agent has to find a key in order to open a door that leads to the next stage. Thus, finding the key should be considered as a subgoal and discovered before reaching the door. It remains for future work to draw inspiration for the available algorithms and test similar approaches in order to investigate more the role of intrinsic motivation and how it develops.

Planning, or cognitive search, is a very complex process. One of the drawbacks in such a process is that there is no behavioral evidence of the planning activity. While model-free processes seem to resemble an online composite process (acting and learning), model-based processes contain an offline phase in which the simulation of a decision tree takes place. This phase is not linked directly to behavior and only implicitly can be linked as in our case. The planning process is a simulation process and it is very difficult to identify behavioral patterns of planning activity, and incorporate this information into the modeling process. One possibility is that there might be a lot of information about the actual planning process contained in eye-movement behavior. It is very likely that modeling eye movement control

could directly link the simulated activity that takes place during planning with behavioral evidence.

To conclude, we would like to close with a comment on a future direction that machine learning methods, applied in Cognitive Neuroscience, can take: to use the model parameters to cluster participants (or even patients). Computerized version of experiments can provide us with the ability to test a large number of people, and not be restrained by the limited number of people in a school or invited to a lab. Smartphone applications such as the Great Brain Experiment (Brown et al., 2014), and Amazon's Mechanical Turk (Buhrmester et al., 2011) can provide a form of crowdsourcing, which can allow researchers to test a large population of subjects. With such information, we would be in a position in which we could use the inferred model parameters as a criterion for clustering subjects according to their behavior. We suggest that these approaches might help also to identify underlying medical conditions such as schizophrenia, ADHD, autism, etc.

7.4 Final Words

The planning mechanism(s) underlying the brain's decision making processes has been of major concern for decades and still remains elusive. Model-based RL provides a theoretically elegant framework to explain human behavior. Although a problem can be translated into an MDP and then solved by a model-based algorithm, the human planning system seems to be affected by perceptual features of that problem. For example, in the ToL problem, we saw that a participant is affected by a similarity measure of his or her current state with the goal state. One possible direction is to integrate Bayesian models with model-based tree search methods, in order to understand how characteristics of the task and perception affects planning. Achieving this might help to understand how the perceptual system affects the planning process of neurologically impaired patients and suggest solutions to improve their lives. We showed how to address challenges in decision making which involve eye-tracking by utilizing RL techniques. Perhaps one of the most exciting directions to develop further within our framework is to integrate the information taken from eye-tracking data with behavioral models. We hope that the computational approaches and ideas described here will be utilized within the developmental community and extended to mathematical psychology theories.

References

Albert, D. and Steinberg, L. (2011). Age differences in strategic planning as indexed by the tower of london. *Child Development*, 82(5):1501–1517.

Anderson, P., Anderson, V., and Lajoie, G. (1996). The tower of london test: Validation and standardization for pediatric populatons. *The Clinical Neuropsychologist*, 10(1):54–65.

Aslin, R. N. (2007). What's in a look? *Developmental Science*, 10(1):48–53.

Aslin, R. N., Saffran, J. R., and Newport, E. L. (1998). Computation of conditional probability statistics by 8-month-old infants. *Psychological Science*, 9(4):321–324.

Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256.

Badre, D., Doll, B. B., Long, N. M., and Frank, M. J. (2012). Rostrolateral prefrontal cortex and individual differences in uncertainty-driven exploration. *Neuron*, 73(3):595–607.

Bakker, B. and Schmidhuber, J. (2004). Hierarchical reinforcement learning based on subgoal discovery and subpolicy specialization. In *Proc. of the 8-th Conf. on Intelligent Autonomous Systems*, pages 438–445.

Balaguer, J., Spiers, H., Hassabis, D., and Summerfield, C. (2016). Neural mechanisms of hierarchical planning in a virtual subway network. *Neuron*, 90(4):893–903.

Baldwin, D., Andersson, A., Saffran, J., and Meyer, M. (2008). Segmenting dynamic human action via statistical structure. *Cognition*, 106(3):1382–1407.

Balleine, B. W., Daw, N. D., and O'Doherty, J. P. (2008). Multiple forms of value learning and the function of dopamine. *Neuroeconomics: decision making and the brain*, 36:7–385.

Balleine, B. W. and Dickinson, A. (1998). Goal-directed instrumental action: contingency and incentive learning and their cortical substrates. *Neuropharmacology*, 37(4):407–419.

Balleine, B. W. and O'Doherty, J. P. (2010). Human and rodent homologies in action control: corticostriatal determinants of goal-directed and habitual action. *Neuropsychopharmacology*, 35(1):48–69.

Barto, A., Sutton, R., and Anderson, C. (1983). Neuronlike adaptive elements that can solve difficult learning control problems. *Systems, Man and Cybernetics, IEEE Transactions on*, SMC-13(5):834–846.

Baughman, F. D. and Cooper, R. P. (2007). Inhibition and young children's performance on the tower of london task. *Cognitive Systems Research*, 8(3):216 – 226. Cognitive Modeling.

Bellemare, M. G., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., and Munos, R. (2016). Unifying count-based exploration and intrinsic motivation. *arXiv preprint arXiv:1606.01868*.

Bellman, R. (1954). The theory of dynamic programming. *Bull. Amer. Math. Soc.*, 60(6):503–515.

Bertsekas, D. P. (2000). *Dynamic Programming and Optimal Control*. Athena Scientific, 2nd edition.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Boden, M. A. (1988). *Computer models of mind: Computational approaches in theoretical psychology*. Cambridge University Press.

Bonawitz, E. B., van Schijndel, T. J., Friel, D., and Schulz, L. (2012). Children balance theories and evidence in exploration, explanation, and learning. *Cognitive psychology*, 64(4):215–234.

Botvinick, M. M., Niv, Y., and Barto, A. C. (2009). Hierarchically organized behavior and its neural foundations: a reinforcement learning perspective. *Cognition*, 113(3):262–80.

Brainard, D. H. (1997). The psychophysics toolbox. *Spatial vision*, 10:433–436.

Branavan, S., Silver, D., and Barzilay, R. (2011). Non-linear monte-carlo search in civilization ii. AAAI Press/International Joint Conferences on Artificial Intelligence.

Brown, H. R., Zeidman, P., Smittenaar, P., Adams, R. A., McNab, F., Rutledge, R. B., and Dolan, R. J. (2014). Crowdsourcing for cognitive science–the utility of smartphones. *PloS one*, 9(7):e100662.

Bruner, J. S. (1973). Organization of early skilled action. *Child development*, 44(1):1–11.

Buchsbaum, D., Griffiths, T. L., Plunkett, D., Gopnik, A., and Baldwin, D. (2015). Inferring action structure and causal relationships in continuous sequences of human action. *Cognitive psychology*, 76:30–77.

Buhrmester, M., Kwang, T., and Gosling, S. D. (2011). Amazon's mechanical turk a new source of inexpensive, yet high-quality, data? *Perspectives on psychological science*, 6(1):3–5.

Bull, R., Espy, K. A., and Senn, T. E. (2004). A comparison of performance on the towers of london and hanoi in young children. *Journal of Child Psychology and Psychiatry*, 45(4):743–754.

Butko, N. J. and Movellan, J. R. (2010). Infomax control of eye movements. *Autonomous Mental Development, IEEE Transactions on*, 2(2):91–107.

Caligiore, D., Borghi, A. M., Parisi, D., and Baldassarre, G. (2010). TRoPICALS: a computational embodied neuroscience model of compatibility effects. *Psychological review*, 117(4):1188–228.

Camerer, C. and Hua Ho, T. (1999). Experience-weighted attraction learning in normal form games. *Econometrica*, 67(4):827–874.

Cassandra, A. R., Kaelbling, L. P., and Littman, M. L. (1994). Acting optimally in partially observable stochastic domains. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (Vol. 2)*, AAAI'94, pages 1023–1028, Menlo Park, CA, USA. American Association for Artificial Intelligence.

Cavanagh, J. F., Figueroa, C. M., Cohen, M. X., and Frank, M. J. (2011). Frontal theta reflects uncertainty and unexpectedness during exploration and exploitation. *Cerebral cortex*, page bhr332.

Chentanez, N., Barto, A. G., and Singh, S. P. (2005). Intrinsically Motivated Reinforcement Learning. *Advances in Neural Information Processing Systems*, pages 1281–1288.

Christakou, A., Gershman, S. J., Niv, Y., Simmons, A., Brammer, M., and Rubia, K. (2013). Neural and psychological maturation of decision-making in adolescence and young adulthood. *Journal of cognitive neuroscience*, 25(11):1807–1823.

Cooper, R. and Shallice, T. (2000). Contention scheduling and the control of routine activities. *Cognitive neuropsychology*, 17(4):297–338.

Courville, A. C., Daw, N. D., and Touretzky, D. S. (2006). Bayesian theories of conditioning in a changing world. *Trends in cognitive sciences*, 10(7):294–300.

Daw, N. D. (2011). Trial-by-trial data analysis using computational models. Technical report.

Daw, N. D. (2012). Model-based reinforcement learning as cognitive search: neurocomputational theories. *Cognitive search: Evolution, algorithms and the brain*, pages 195–208.

Daw, N. D., Courville, A. C., Tourtezky, D. S., and Touretzky, D. S. (2006). Representation and timing in theories of the dopamine system. *Neural computation*, 18(7):1637–77.

Daw, N. D., Gershman, S. J., Seymour, B., Dayan, P., and Dolan, R. J. (2011). Model-based influences on humans' choices and striatal prediction errors. *Neuron*, 69(6):1204–1215.

Daw, N. D., Niv, Y., and Dayan, P. (2005). Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control. *Nature neuroscience*, 8(12):1704–1711.

Daw, N. D., Tobler, P. N., Glimcher, P., and Fehr, E. (2013). Value learning through reinforcement: the basics of dopamine and reinforcement learning. *Neuroeconomics,*, pages 283–298.

Dayan, P. and Abbott, L. F. (2005). *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. The MIT Press.

Dayan, P. and Daw, N. D. (2008). Decision theory, reinforcement learning, and the brain. *Cognitive, Affective, & Behavioral Neuroscience*, 8(4):429–453.

Dayan, P. and Jyu, A. (2003). Uncertainty and learning. *IETE Journal of Research*, 49(2-3):171–181.

Dearden, R., Friedman, N., and Andre, D. (1999). Model based bayesian exploration. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI'99, pages 150–159, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Dearden, R., Friedman, N., and Russell, S. (1998). Bayesian q-learning. In *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, AAAI '98/IAAI '98, pages 761–768, Menlo Park, CA, USA. American Association for Artificial Intelligence.

Deci, E. L. and Ryan, R. M. (1985). *Intrinsic Motivation and Self-Determination in Human Behavior*. Springer Science & Business Media.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Dickinson, A. (1985). Actions and Habits: The Development of Behavioural Autonomy. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 308(1135):67–78.

Dietterich, T. G. (1998). The maxq method for hierarchical reinforcement learning. In *In Proceedings of the Fifteenth International Conference on Machine Learning*, pages 118–126. Morgan Kaufmann.

Dolan, R. J. and Dayan, P. (2013). Goals and habits in the brain. *Neuron*, 80(2):312–325.

Doll, B. B., Jacobs, W. J., Sanfey, A. G., and Frank, M. J. (2009). Instructional control of reinforcement learning: A behavioral and neurocomputational investigation. *Brain Research*, 1299:74 – 94.

Doya, K. (1999). What are the computations of the cerebellum, the basal ganglia and the cerebral cortex? *Neural Networks.*, 12(7-8):961–974.

Duff, M. O. (2002). *Optimal Learning: Computational Procedures for Bayes-adaptive Markov Decision Processes*. PhD thesis. AAI3039353.

Fernández, F., García, J., and Veloso, M. (2010). Probabilistic policy reuse for inter-task transfer learning. *Robotics and Autonomous Systems*, 58(7):866–871.

Fischer, K. W. (1980). A Theory of Cognitive Development: The Control and Construction of Hierarchies of Skills. *Psychological Review*, 87(6):477–531.

Foster, D. J. and Wilson, M. A. (2007). Hippocampal theta sequences. *Hippocampus*, 17(11).

Frank, M. J. (2007). Go and nogo learning and the basal ganglia. http://www.dana.org/Cerebrum/Default.aspx?id=39393.

Frank, M. J. and Claus, E. D. (2006). Anatomy of a decision: striato-orbitofrontal interactions in reinforcement learning, decision making, and reversal. *Psychological review*, 113(2):300.

Frank, M. J., Doll, B. B., Oas-Terpstra, J., and Moreno, F. (2009). The neurogenetics of exploration and exploitation: Prefrontal and striatal dopaminergic components. *Nature neuroscience*, 12(8):1062.

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

Gelly, S. and Silver, D. (2011). Monte-carlo tree search and rapid action value estimation in computer go. *Artificial Intelligence*, 175(11):1856–1875.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2014). *Bayesian data analysis*, volume 2. Taylor & Francis.

Gershman, S. J., Cohen, J. D., and Niv, Y. (2010). Learning to selectively attend. In *32nd Annual Conference of the Cognitive Science Society*.

Gershman, S. J., Markman, A. B., and Otto, A. R. (2014). Retrospective revaluation in sequential decision making: A tale of two systems. *Journal of Experimental Psychology: General*, 143(1):182.

Gibson, J. J. (1986). *The Ecological Approach to Visual Perception*. Routledge.

Gilhooly, K. J., Phillips, L. H., Wynn, V., Logie, R. H., and Della Sala, S. (1999). Planning processes and age in the five-disc tower of london task. *Thinking & reasoning*, 5(4):339–361.

Gläscher, J., Daw, N., Dayan, P., and O'Doherty, J. P. (2010). States versus rewards: dissociable neural prediction error signals underlying model-based and model-free reinforcement learning. *Neuron*, 66(4):585–595.

Goel, V. and Grafman, J. (1995). Are the frontal lobes implicated in 'planning' functions? interpreting data from the tower of hanoi. *Neuropsychologia*, 33(5):623 – 642.

Goel, V., Pullara, D., and Grafman, J. (2001). A computational model of frontal lobe dysfunction: working memory and the tower of hanoi task. *Cognitive Science*, 25(2):287–313.

Guez, A. (2015). *Sample-based Search Methods for Bayes-Adaptive Planning*. Ph.D. thesis, Gatsby computational neuroscience unit, University College London.

Guez, A., Heess, N., Silver, D., and Dayan, P. (2014a). Bayes-adaptive simulation-based search with value function approximation. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., editors, *Advances in Neural Information Processing Systems 27*, pages 451–459. Curran Associates, Inc.

Guez, A., Niyogi, R., Bach, D., Guitart-Massip, M., Dolan, R., and Dayan, P. (2013a). A normative theory of approach-avoidance conflicts during dynamic foraging in humans. In *The 1st Multidisciplinary Conference on Reinforcement Learning and Decision Making (RLDM), Pinceton University, New Jersey, United States*.

Guez, A., Silver, A., and Dayan, P. (2012). Efficient Bayes-Adaptive Reinforcement Learning using Sample-Based Search.

Guez, A., Silver, D., and Dayan, P. (2013b). Scalable and efficient bayes-adaptive reinforcement learning based on monte-carlo tree search. *Journal of Artificial Intelligence Research*, pages 841–883.

Guez, A., Silver, D., and Dayan, P. (2014b). Better optimism by bayes: Adaptive planning with rich models. *arXiv preprint arXiv:1402.1958*.

Harlow, H. F. (1950). Learning and satiation of response in intrinsically motivated complex puzzle performance by monkeys. *Journal of comparative and physiological psychology*, 43(4):289–294.

Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109.

Hayes, T. R., Petrov, A. A., and Sederberg, P. B. (2011). A novel method for analyzing sequential eye movements reveals strategic influence on raven's advanced progressive matrices. *Journal of Vision*, 11(10):10–10.

Heinrich, J. and Silver, D. (2014). Self-play monte-carlo tree search in computer poker. In *Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence*.

Hodgson, T. L., Bajwa, A., Owen, A. M., and Kennard, C. (2000). The strategic control of gaze direction in the tower of london task. *Journal of Cognitive Neuroscience*, 12(5):894–907.

Hula, A., Montague, P. R., and Dayan, P. (2015). Monte carlo planning method estimates planning horizons during interactive social exchange. *PLoS Comput Biol*, 11(6).

Huys, Q. J. M., Eshel, N., O'Nions, E., Sheridan, L., Dayan, P., and Roiser, J. P. (2012). Bonsai trees in your head: how the pavlovian system sculpts goal-directed choices by pruning decision trees. *PLoS computational biology*, 8(3):e1002410.

Huys, Q. J. M., Lally, N., Faulkner, P., Eshel, N., Seifritz, E., Gershman, S. J., Dayan, P., and Roiser, J. P. (2015). Interplay of approximate planning strategies. *Proceedings of the National Academy of Sciences*, 112(10):3098–3103.

Just, M. A. and Carpenter, P. A. (1992). A capacity theory of comprehension: Individual differences in working memory. *Psychological Review*, pages 122–149.

Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artif. Intell.*, 101(1-2):99–134.

Kaller, C. P., Rahm, B., Bolkenius, K., and Unterrainer, J. M. (2009). Eye movements and visuospatial problem solving: Identifying separable phases of complex cognition. *Psychophysiology*, 46(4):818–830.

Kaller, C. P., Rahm, B., Spreer, J., Mader, I., and Unterrainer, J. M. (2008). Thinking around the corner: The development of planning abilities. *Brain and Cognition*, 67(3):360 – 370.

Kaller, C. P., Unterrainer, J. M., Rahm, B., and Halsband, U. (2004). The impact of problem structure on planning: insights from the tower of london task. *Cognitive Brain Research*, 20(3):462 – 472.

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

Kazemitabar, S. J. and Beigy, H. (2009). *Advances in Neuro-Information Processing*, volume 5506 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg.

Kearns, M., Mansour, Y., and Ng, A. Y. (2002). A sparse sampling algorithm for near-optimal planning in large markov decision processes. *Mach. Learn.*, 49(2-3):193–208.

Kenward, B., Folke, S., Holmberg, J., Johansson, A., and Gredebäck, G. (2009). Goal directedness and decision making in infants. *Developmental psychology*, 45(3):809.

Kidd, C., Piantadosi, S. T., and Aslin, R. N. (2012). The goldilocks effect: Human infants allocate attention to visual sequences that are neither too simple nor too complex. *PloS one*, 7(5):e36399.

Kidd, C., Piantadosi, S. T., and Aslin, R. N. (2014). The goldilocks effect in infant auditory attention. *Child Development*, 85(5):1795–1804.

Kimura, A., Pang, D., Takeuchi, T., Miyazato, K., Yamato, J., and Kashino, K. (2010). A stochastic model of human visual attention with a dynamic bayesian network. *submitted, IEEE Trans. Pattern Anal. Mach. Intell.*

Klahr, D. and Robinson, M. (1981). Formal assessment of problem-solving and planning processes in preschool children. *Cognitive Psychology*, 13(1):113–148.

Kleiner, M., Brainard, D., Pelli, D., Ingling, A., Murray, R., Broussard, C., et al. (2007). Whatńew in psychtoolbox-3. *Perception*, 36(14):1.

Klossek, U., Russell, J., and Dickinson, A. (2008). The control of instrumental action following outcome devaluation in young children aged between 1 and 4 years. *Journal of Experimental Psychology: General*, 137(1):39.

Klossek, U. M., Yu, S., and Dickinson, A. (2011). Choice and goal-directed behavior in preschool children. *Learning & behavior*, 39(4):350–357.

Kocsis, L. and Szepesvári, C. (2006). Bandit based monte-carlo planning. In *Proceedings of the 17th European Conference on Machine Learning*, ECML'06, pages 282–293, Berlin, Heidelberg. Springer-Verlag.

Kotovsky, K., Hayes, J. R., and Simon, H. A. (1985). Why are some problems hard? evidence from tower of hanoi. *Cognitive psychology*, 17(2):248–294.

Kulkarni, T. D., Narasimhan, K. R., Saeedi, A., and Tenenbaum, J. B. (2016). Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. *arXiv preprint arXiv:1604.06057*.

Lashley, K. S. (1951). The problem of serial order in behavior. pages 112–136.

Le Pelley, M. E. (2004). The role of associative history in models of associative learning: A selective review and a hybrid model. *Quarterly Journal of Experimental Psychology Section B*, 57(3):193–243.

Lee, D., Seo, H., and Jung, M. W. (2012). Neural basis of reinforcement learning and decision making. *Annual review of neuroscience*, 35:287.

Littman, M. L. (1996). *Algorithms for Sequential Decision Making*. PhD thesis, Brown University.

Lucas, É. (1882). *Récréations mathématiques*, volume 1. Gauthier-Villars.

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

Mackintosh, N. J. (1975). A theory of attention: variations in the associability of stimuli with reinforcement. *Psychological review*, 82(4):276.

Marin, J.-M., Pudlo, P., Robert, C. P., and Ryder, R. J. (2012). Approximate bayesian computational methods. *Statistics and Computing*, 22(6):1167–1180.

MATLAB (2015). *version 8.6.0 (R2015b)*. The MathWorks Inc., Natick, Massachusetts.

McGovern, A. and Barto, A. G. (2001). Automatic discovery of subgoals in reinforcement learning using diverse density. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 361–368. Morgan Kaufmann Publishers Inc.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092.

Meyer, M., Baldwin, D. A., and Sage, K. (2011). Assessing young children's hierarchical action segmentation. In *Proceedings of the 33rd annual conference of the cognitive science society*, pages 3156–3161. Cognitive Science Society Boston, MA.

Meyer, M., DeCamp, P., Hard, B., Baldwin, D., and Roy, D. (2010). Assessing behavioral and computational approaches to naturalistic action segmentation. In *Proc. of the 33nd Annual Conference of the Cognitive Science Society*. Citeseer.

Monroy, C., Gerson, S. A., and Hunnius, S. (2015a). Action prediction based on statistical learning. (Manuscript under review).

Monroy, C., Gerson, S. A., and Hunnius, S. (2015b). Infant's action prediction: Statistical learning of continuous action sequences. (Manuscript under review).

Montague, P. R., Dayan, P., and Sejnowski, T. J. (1996). A framework for mesencephalic dopamine systems based on predictive Hebbian learning. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 16(5):1936–47.

Moustafa, A. A., Cohen, M. X., Sherman, S. J., and Frank, M. J. (2008). A role for dopamine in temporal decision making and reward maximization in parkinsonism. *The Journal of Neuroscience*, 28(47):12294–12304.

Najemnik, J. and Geisler, W. S. (2005). Optimal eye movement strategies in visual search. *Nature*, 434(7031):387–391.

Newman, S. D., Carpenter, P. A., Varma, S., and Just, M. A. (2003). Frontal and parietal participation in problem solving in the tower of london: fmri and computational modeling of planning and high-level perception. *Neuropsychologia*, 41(12):1668 – 1682.

Ng, A. Y., Harada, D., and Russell, S. J. (1999). Policy Invariance Under Reward Transformations: Theory and Application to Reward Shaping. *International Conference on Machine Learning*, 16:278–287.

Niv, Y. (2009). Reinforcement learning in the brain. *Journal of Mathematical Psychology*, 53(3):139–154.

Norman, D. and Shallice, T. (1986). Attention to Action : Willed and Automatic Control of Behavior Technical Report No. 8006 / Donald A. Norman and Tim Shallice. [microform] : - Version details - Trove.

Paquet, S., Tobin, L., and Chaib-draa, B. (2005). An online POMDP algorithm for complex multiagent environments. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems - AAMAS '05*, page 970, New York, New York, USA. ACM Press.

Parrila, R. K., Das, J., and Dash, U. N. (1996). Development of planning and its relation to other cognitive processes. *Journal of Applied Developmental Psychology*, 17(4):597–624.

Pearce, J. M. and Hall, G. (1980). A model for pavlovian learning: variations in the effectiveness of conditioned but not of unconditioned stimuli. *Psychological review*, 87(6):532.

Pelli, D. G. (1997). The videotoolbox software for visual psychophysics: Transforming numbers into movies. *Spatial vision*, 10(4):437–442.

Pennington, B. F., Bennetto, L., McAleer, O., and Roberts Jr, R. J. (1996). Executive functions and working memory: Theoretical and measurement issues. *Attention, memory, and executive function*, pages 327–348.

Pfeiffer, B. E. and Foster, D. J. (2013). Hippocampal place-cell sequences depict future paths to remembered goals. *Nature*, 497(7447):74–79.

Phillips, L. H., Wynn, V. E., McPherson, S., and Gilhooly, K. J. (2001). Mental planning and the Tower of London task. *The Quarterly journal of experimental psychology. A, Human experimental psychology*, 54(2):579–97.

Pineau, J., Gordon, G., and Thrun, S. (2003). Point-based value iteration: an anytime algorithm for POMDPs. pages 1025–1030.

Puterman, M. L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc.

Rescorla, R. (1966). Predictability and number of pairings in pavlovian fear conditioning. *Psychonomic Science*, 4(11):383–384.

Rescorla, R. A., Wagner, A. R., et al. (1972). A theory of pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. *Classical conditioning II: Current research and theory*, 2:64–99.

Rosa, M. (2012). *Development and application of model selection methods for investigating brain function*. Ph.D. thesis, University College London.

Ross, S. (2007). Bayes-adaptive pomdps: Toward an optimal policy for learning pomdps with parameter uncertainty. *Course Project report*.

Ross, S. and Chaib-Draa, B. (2007). Aems: An anytime online search algorithm for approximate policy refinement in large pomdps. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, IJCAI'07, pages 2592–2598, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Rummery, G. A. and Niranjan, M. (1994). *On-line Q-learning using connectionist systems*. University of Cambridge, Department of Engineering.

Saffran, J. R., Aslin, R. N., and Newport, E. L. (1996). Statistical learning by 8-month-old infants. *Science*, 274(5294):1926–1928.

Satia, J. and Lave, R. (1973). Markovian decision processes with probabilistic observation of states. *Management Science*, 20(1):1–13.

Schultz, W. (1998). Predictive reward signal of dopamine neurons. *Journal of neurophysiology*, 80(1):1–27.

Schultz, W., Dayan, P., and Montague, P. R. (1997). A neural substrate of prediction and reward. *Science (New York, N.Y.)*, 275(5306):1593–9.

Schulz, L. E. and Bonawitz, E. B. (2007). Serious fun: preschoolers engage in more exploratory play when evidence is confounded. *Developmental psychology*, 43(4):1045.

Shallice, T. (1982). Specific impairments of planning. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 298(1089):199–209.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489.

Silver, D., Sutton, R. S., and Müller, M. (2008). Sample-based learning and search with permanent and transient memories. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 968–975, New York, NY, USA. ACM.

Simon, H. A. (1975). The functional equivalence of problem solving skills. *Cognitive Psychology*, 7(2):268–288.

Smith, T. and Simmons, R. (2004). Heuristic search value iteration for pomdps. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI '04, pages 520–527, Arlington, Virginia, United States. AUAI Press.

Smith, T. and Simmons, R. (2005). Point-based POMDP Algorithms: Improved Analysis and Implementation. In *Proc. of the Conference on Uncertainty in Artificial Intelligence*.

Spaan, M. T. and Vlassis, N. (2005). Perseus: Randomized point-based value iteration for pomdps. *Journal of artificial intelligence research*, 24:195–220.

Stephan, K. E., Penny, W. D., Daunizeau, J., Moran, R. J., and Friston, K. J. (2009). Bayesian model selection for group studies. *Neuroimage*, 46(4):1004–1017.

Strens, M. (2000). A bayesian framework for reinforcement learning. In *In Proceedings of the Seventeenth International Conference on Machine Learning*, pages 943–950. ICML.

Sutton, R. and Barto, A. (1998). *Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning)*. A Bradford Book.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44.

Sutton, R. S., Precup, D., and Singh, S. (1999). Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1):181–211.

Szepesvári, C. (2010). Algorithms for reinforcement learning. *Synthesis lectures on artificial intelligence and machine learning*, 4(1):1–103.

Tanaka, S. C., Samejima, K., Okada, G., Ueda, K., Okamoto, Y., Yamawaki, S., and Doya, K. (2006). Brain mechanism of reward prediction under predictable and unpredictable environmental dynamics. *Neural Networks*, 19(8):1233–1241.

Tummeltshammer, K. S. and Kirkham, N. Z. (2013). Learning to look: probabilistic variation and noise guide infants' eye movements. *Developmental science*, 16(5):760–771.

Unterrainer, J., Kaller, C., Halsband, U., and Rahm, B. (2006). Planning abilities and chess: A comparison of chess and non-chess players on the tower of london task. *British Journal of Psychology*, 97(3):299–311.

van Casteren, M. and Davis, M. H. (2006). Mix, a program for pseudorandomization. *Behavior research methods*, 38(4):584–589.

Waldau, R. (1999). A developmental study of problem solving in children aged 3–6: Development of planning strategies. Undergraduate disertation.

Washington, R. (1997). Bi-pomdp: Bounded, incremental, partially-observable markov-model planning. In *Proceedings of the 4th European Conference on Planning: Recent Advances in AI Planning*, ECP '97, pages 440–451, London, UK, UK. Springer-Verlag.

Watkins, C. J. C. H. (1989). *Learning from Delayed Rewards*. PhD thesis, King's College, Cambridge, UK.

Watkins, C. J. C. H. and Dayan, P. (1992). Technical note: Q-learning. *Mach. Learn.*, 8(3-4):279–292.

Welsh, M. C., Satterlee-Cartmell, T., and Stine, M. (1999). Towers of hanoi and london: Contribution of working memory and inhibition to performance. *Brain and Cognition*, 41(2):231 – 242.

Wilson, M. A. and Foster, D. J. (2006). Reverse replay of behavioural sequences in hippocampal place cells during the awake state. *Nature*, 440:680–683.

Wood, W. and Neal, D. T. (2007). A new look at habits and the habit-goal interface. *Psychological review*, 114(4):843–63.

Wunderlich, K., Smittenaar, P., and Dolan, R. J. (2012). Dopamine enhances model-based over model-free choice behavior. *Neuron*, 75(3):418–424.